IPv6 Ready Logo Phase-2

Conformance Test Specification
IKEv2

Technical Document

Revision 1.0.1

IPv6 Forum *http://www.ipv6forum.org/*
IPv6 Ready Logo Committee *http://www.ipv6ready.org/*

MODIFICATION RECORD

Version 1.0.1 Apr. 15, 2009

- IKEv2.EN.I.1.1.5.2, IKEv2.SGW.1.1.5.2, IKEv2.EN.R.1.1.5.3, IKEv2.SGW.R.1.1.5.3, IKEv2.EN.R.1.1.5.4, IKEv2.SGW.R.1.1.5.4 - Update acceptable packets and check establishment of IKE_SA
- IKEv2.EN.I.1.1.5.3, IKEv2.SGW.I.1.1.5.3 - Add new test cases for Intetaction of COOKIE and INVALID_KE_PAYLOAD with unoptimized Responder

Version 1.0.0 Dec. 11, 2008

- Initial release

ACKNOWLEDGMENTS

The IPv6 Forum would like to acknowledge the efforts of the following organizations in the development of this test suite.

Authors:

Yokogawa Electric Corporation
Nippon Telegraph and Telephone Corporation (NTT)

Commentators:

NTT Advanced Technology Corporation (NTT-AT)

Note:

Development of this document was supported in part by a grant from NICT.

INTRODUCTION

Overview

TAHI Project is the joint effort formed with the objective of developing and providing the verification technology for IPv6.

The growth process of IPv4 was the history of encountering various kinds of obstacles and conquering such obstacles. However, once the position as infrastructure was established, it is not allowed to repeat the same history.
This is a reason why the verification technology is essential for IPv6 deployment.

> We research and develop conformance tests and interoperability tests for IPv6.
>
> We closely work with the KAME project and USAGI project.
> We help activities of these projects in the quality side by offering the verification technology we develop in TAHI project and improve the development efficiency.
>
> We open the results and fruits of the project to the public for FREE.
> Any developer concerned with IPv6 can utilize the results and fruits of TAHI project freely.
> Free software plays an important role in progress of the Internet. We believe that providing the verification technology for FREE contributes to advances of IPv6.
> Besides the programs, the specifications and criteria of verification will be included in the Package.

Abbreviations and Acronyms

TN: Testing Node
TH: Testing Host
TR: Testing Router
NUT: Node Under Test
HUT: Host Under Test
RUT: Router Under Test
IKE: Internet Key Exchange (IKEv2) Protocol
EN: End-Node
SGW: Security-Gateway
PSK: Pre-Shared Key
AUTH: Authentication Payload
CERT: Certificate Payload
CERTREQ: Certificate Request Payload
CP: Configuration Payload
D: Delete Payload
E: Encrypted Payload
EAP: Extensible Authentication Payload
HDR: IKE Header
IDi: Identification - Initiator Payload
IDr: Identification - Responder Payload
KE: Key Exchange Payload
Ni: Nonce - Initiator Payload
Nr: Nonce - Responder Payload
N: Notify Payload
SA: Security Association Payload
TSi: Traffic Selector - Initiator Payload

TSr: Traffic Selector - Responder Payload
V: Vendor ID Payload

TEST ORGANIZATION

This document organizes tests by Section based on related test methodology or goals. Each group begins with a brief set of comments pertaining to all tests within that group. This is followed by a series of description blocks; each block describes a single test. The format of the description block is as follows:

Test Label:	The test label and title comprise the first line of the test block. The test label is composed by concatenating the short test suite name, the section number, the group number, and the test number within the group. These elements are separated by periods. The Test Number is the section, group and test number, also separated by periods.
Purpose:	The Purpose is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
References:	The References section lists cross-references to the specifications and documentation that might be helpful in understanding and evaluating the test and results.
Resource Requirements:	The Resource Requirements section specifies the software, hardware, and test equipment that will be needed to perform the test.
Test Setup:	The Test Setup section describes the configuration of all devices prior to the start of the test. Different parts of the procedure may involve configuration steps that deviate from what is given in the test setup. If a value is not provided for a protocol parameter, then the protocol's default is used for that parameter.
Procedure:	This section of the test description contains the step-by-step instructions for carrying out the test. These steps include such things as enabling interfaces, unplugging devices from the network, or sending packets from a test station. The test procedure also cues the tester to make observations, which are interpreted in accordance with the observable results given for that test part.
Observable Results:	This section lists observable results that can be examined by the tester to verify that the NUT is operating properly. When multiple observable results are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail for each test is usually based on how the NUT's behavior compares to the results described in this section.
Possible Problems:	This section contains a description of known issues with the test procedure, which may affect test results in certain situations.

REFERENCES

The following documents are referenced in this text:

- RFC 4306 - Internet Key Exchange (IKEv2) Protocol, December, 2005.
- RFC 4307 - Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2), December, 2005
- RFC 4718 – IKEv2 Clarifications and Implementation Guidelines, October, 2006

TABLE OF CONTENTS

Requirements

To obtain the IPv6 Ready Logo Phase-2 for IKEv2, the Node Under Test (NUT) must satisfy all of the following requirements.

Equipment Type

There are two possibilities for equipment types:

End-Node:

A node who can use IKEv2 (IPsec) only for itself. Host and Router can be an End-Node.

SGW (Security Gateway):

A node who can provide IKEv2 (IPsec tunnel mode) for nodes behind it. Router can be a SGW.

Function List

Basic/Advanced Functionality table

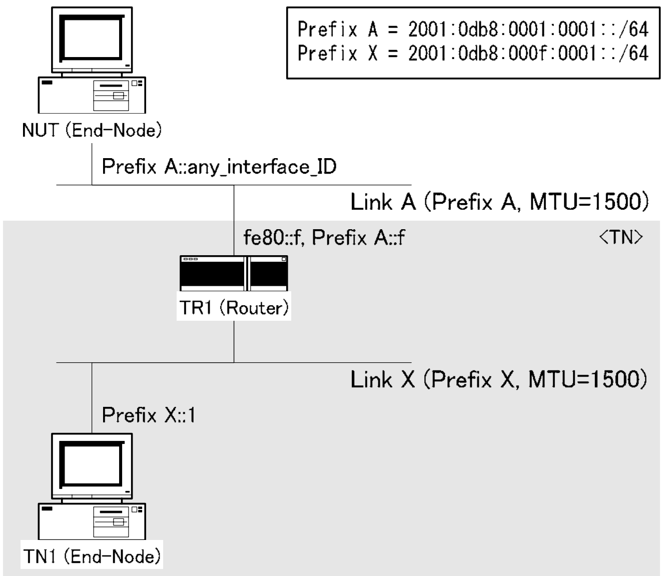
This conformance test specification consists following BASIC/ADVANCED functions.
The tests for ADVANCED functions may be omitted if the NUT does not support the ADVANCED function.
All NUTs are required to support BASIC. ADVANCED is required for all NUTs which support ADVANCED function.

Parameter		BASIC	ADVANCED
Exchange Type		Initial Exchanges (IKE_INIT, IKE_AUTH)	-
		CREATE_CHILD_SA	-
		INFORMATIONAL	-
IKE_SA	Encryption Algorithm	ENCR_3DES	ENCR_AES_CBC ENCR_AES_CTR
	Pseudo-random Function	PRF_HMAC_SHA1	PRF_AES128_XCBC
	Integrity Algorithm	AUTH_HMAC_SHA1_96	AUTH_AES_XCBC_96
	Diffie-Hellman Group	2 (1024 MODP Group)	14 (2048 MODP Group)
CHILD_SA	Encryption Algorithm	ENCR_3DES	ENCR_AES_CBC ENCR_AES_CTR ENCR_NULL
	Integrity Algorithm	AUTH_HMAC_SHA1_96	AUTH_AES_XCBC_96 NONE
	Extended Sequence Numbers	No Extended Sequence Numbers	Extended Sequence Numbers
Authentication Method		PSK	-
Security Protocol		ESP	-
Encapsulation mode	End-Node	Transport	Tunnel
	SGW	Tunnel	-
Multiple Proposals		Receiving	Sending and Receiving
Multiple Transforms		Receiving	Sending and Receiving
Liveness Check		Support	-
Cookies		-	Support

Rekeying	Support	-
Traffic Selector Negotiation	Support	-
Requesting an Internal Address on a Remote Network	-	Support
Perfect Forward Secrecy	-	Support
Closing SAs	Support	

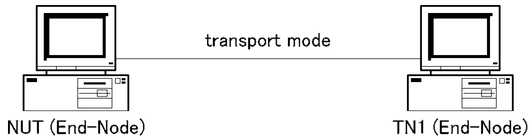
Common Topology

Common Topology for End-Node: End-Node to End-Node

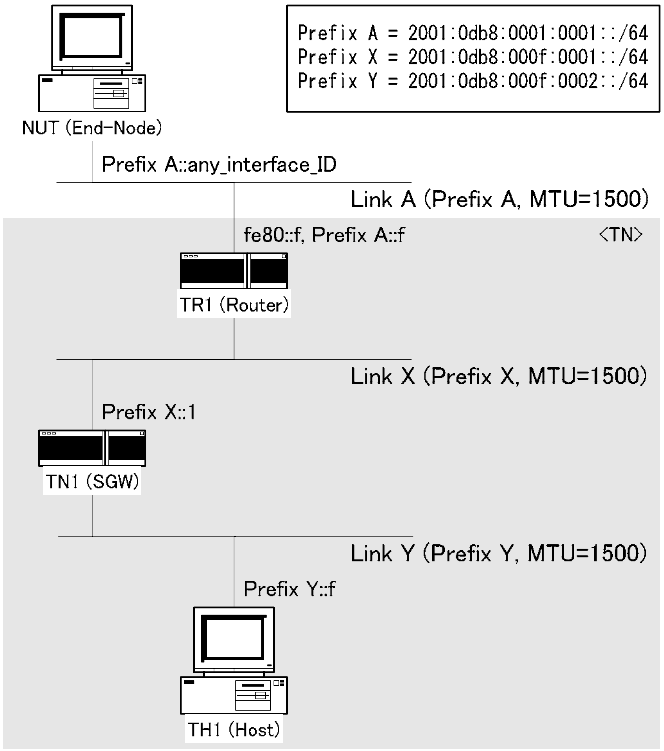


The common topology involves End-Nodes and Router device on each link.

The transport mode is used in this topology.

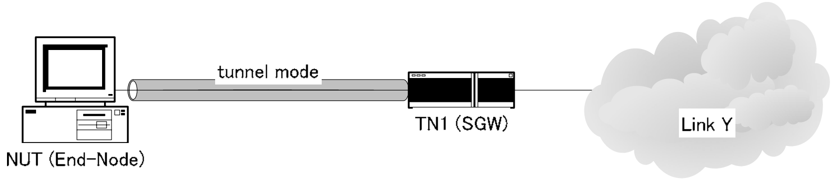


Common Topology for End-Node: End-Node to SGW

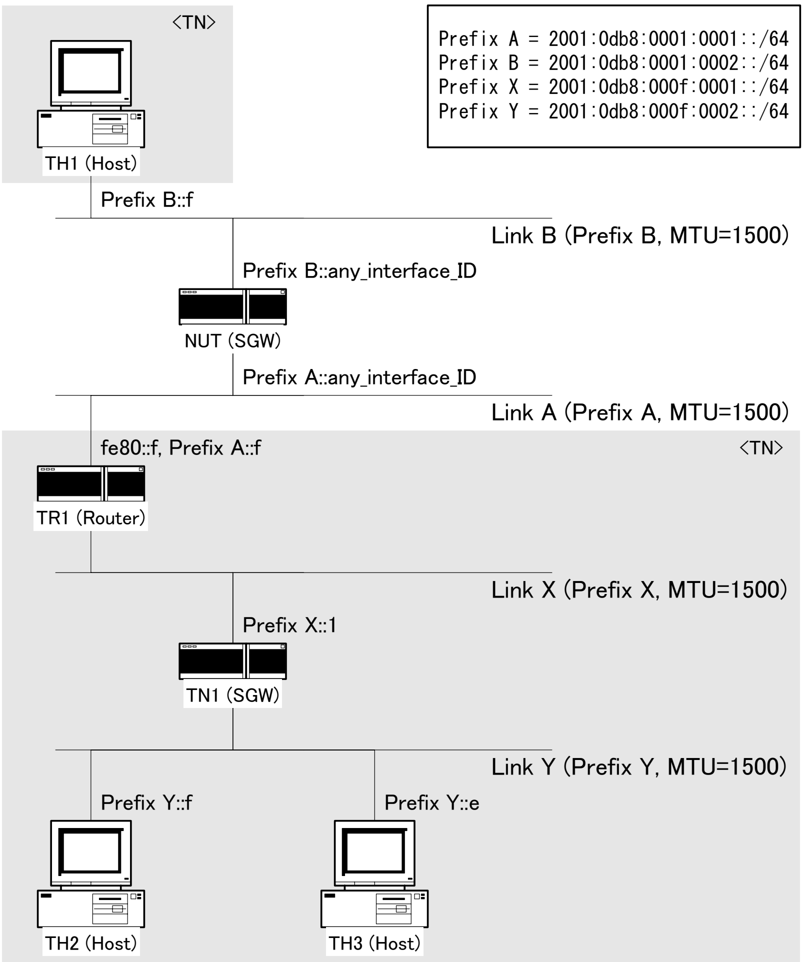


The common topology involves End-Node, SGW and Router device on each link.

The tunnel mode is used in this topology.

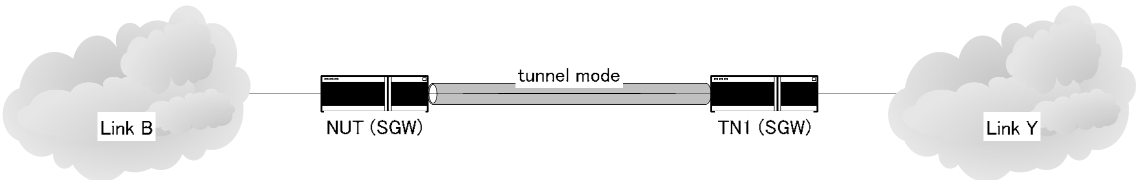


Common Topology for SGW: SGW to SGW

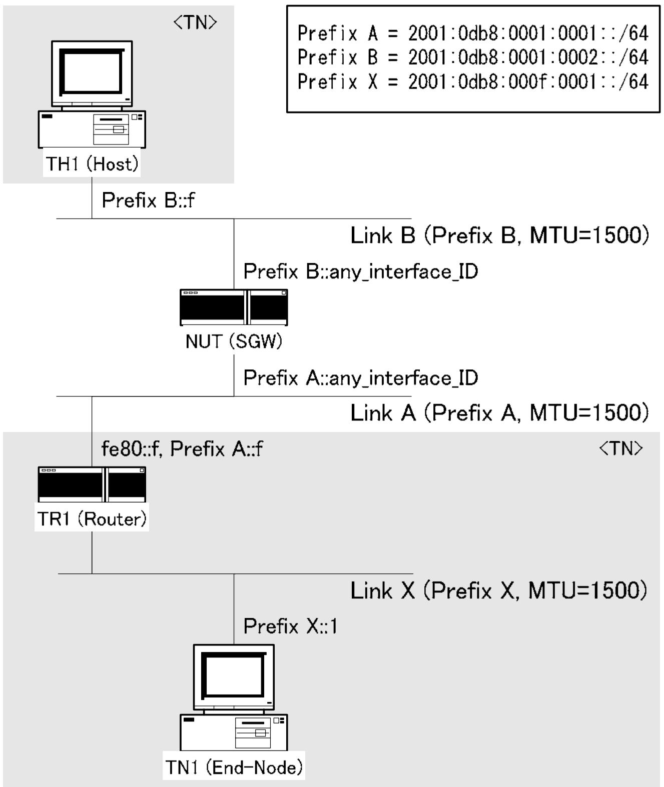


The common topology involves SGWs, Router and Host device on each link.

The tunnel mode is used in this topology.

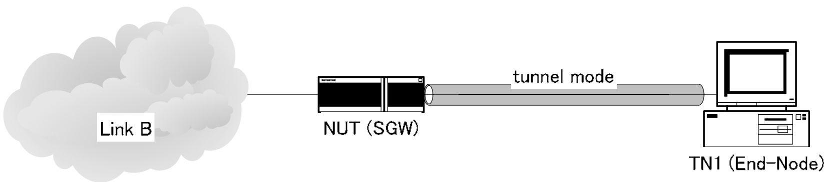


Common Topology for SGW: SGW to End-Node



The common topology involves End-Node, SGW, Router and Host device on each link.

The tunnel mode is used in this topology.



Common Configuration for NUT

Common Configuration for End-Node: End-Node to End-Node

IKE Peer

	Address	Port	Authentication		ID	
			Method	Key Value	Type	Data
Local	NUT	500	PSK	IKETEST12345678!	ID_IPV6_ADDR	NUT
Remote	TN1	500	PSK	IKETEST12345678!	ID_IPV6_ADDR	TN1

IKE_SA

Algorithms			
Encryption	PRF	Integrity	Diffie-Hellman
ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	2 (1024 MODP Group)

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

CHILD_SA

	Security Protocol	Mode	Algorithms		
			Encryption	Integrity	Extended Sequence Numbers
Inbound	ESP	Transport	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Outbound	ESP	Transport	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1	ANY	ANY	NUT	ANY	ANY
Outbound	NUT	ANY	ANY	TN1	ANY	ANY

If NUT is the initiator, NUT must propose Traffic Selector covering above address range.
If NUT is the responder, NUT must narrow Traffic Selector to above address range.

Common Configuration for End-Node: End-Node to SGW

IKE Peer

	Address	Port	Authentication		ID	
			Method	Key Value	Type	Data
Local	NUT	500	PSK	IKETEST123!	ID_IPV6_ADDR	NUT
Remote	TN1 (Link X)	500	PSK	IKETEST456!	ID_IPV6_ADDR	TN1 (LinkX)

IKE_SA

Algorithms			
Encryption	**PRF**	**Integrity**	**Diffie-Hellman**
ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	2 (1024 MODP Group)

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

CHILD_SA

	Security Protocol	Mode	Algorithms		
			Encryption	Integrity	Extended Sequence Numbers
Inbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Outbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT	ANY	ANY
Outbound	NUT	ANY	ANY	Link Y	ANY	ANY

If NUT is the initiator, NUT must propose Traffic Selector covering above address range.
If NUT is the responder, NUT must narrow Traffic Selector to above address range.

Common Configuration for SGW: SGW to SGW

IKE Peer

	Address	Port	Authentication		ID	
			Method	Key Value	Type	Data
Local	NUT (Link A)	500	PSK	IKETEST123!	ID_IPV6_ADDR	NUT (Link A)
Remote	TN1 (Link X)	500	PSK	IKETEST456!	ID_IPV6_ADDR	TN1 (Link X)

IKE_SA

Algorithms			
Encryption	PRF	Integrity	Diffie-Hellman
ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	2 (1024 MODP Group)

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

CHILD_SA

	Security Protocol	Mode	Algorithms		
			Encryption	Integrity	Extended Sequence Numbers
Inbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Outbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	Link Y	ANY	ANY

If NUT is the initiator, NUT must propose Traffic Selector covering above address range.
If NUT is the responder, NUT must narrow Traffic Selector to above address range.

Common Configuration for SGW: SGW to End-Node

IKE Peer

	Address	Port	Authentication		ID	
			Method	Key Value	Type	Data
Local	NUT (Link A)	500	PSK	IKETEST123!	ID_IPV6_ADDR	NUT (Link A)
Remote	TN1	500	PSK	IKETEST456!	ID_IPV6_ADDR	TN1

IKE_SA

Algorithms			
Encryption	**PRF**	**Integrity**	**Diffie-Hellman**
ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	2 (1024 MODP Group)

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

CHILD_SA

	Security Protocol	Mode	Algorithms		
			Encryption	Integrity	Extended Sequence Numbers
Inbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Outbound	ESP	Tunnel	ENCR_3DES	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers

If NUT is the initiator, above proposal must be one of proposals from NUT.
If NUT is the responder, NUT must select above proposal.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1	ANY	ANY

If NUT is the initiator, NUT must propose Traffic Selector covering above address range.
If NUT is the responder, NUT must narrow Traffic Selector to above address range.

Common Packets

Common Packets to be transmitted from Tester are defined as the following tables.Tests in this test specification may refer to these common packets.

IKE_SA_INIT Messages

Common Packet #1 : IKE_SA_INIT request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	Any
	IKE_SA Responder's SPI	0
	Next Payload	33 (SA)
	Major Version	2
	Minor Version	0
	Exchange Type	34 (IKE_SA_INIT)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	1
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
SA Payload	Next Payload	34 (KE)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Table below
KE Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	136
	DH Group #	2
	Reserved	0
	Key Exchange Data	any
Ni, Nr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any

- SA Payload

SA Payload	Next Payload				34 (KE)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		1 (IKE)
			SPI Size		0
			# of Transforms		4
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8

				Transform Type	2 (PRF)
				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

Common Packet #2 : IKE_SA_INIT response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	Any
	Next Payload	33 (SA)
	Major Version	2
	Minor Version	0
	Exchange Type	34 (IKE_SA_INIT)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
SA Payload	Next Payload	34 (KE)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Table below
KE Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	136
	DH Group #	2
	Reserved	0
	Key Exchange Data	any
Ni, Nr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any

- SA Payload

SA Payload	Next Payload				34 (KE)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		1 (IKE)
			SPI Size		0
			# of Transforms		4
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	2 (PRF)
				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)

			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

IKE_AUTH Messages

Common Packet #3 : IKE_AUTH request for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	35 (IKE_AUTH)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	1
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	1
	Length	any
E Payload	Next Payload	35 (IDi)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
IDi Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	24
	ID Type	IPV6_ADDR
	Reserved	0
	Identification Data	TN1's Global Address on Link X
AUTH Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Auth Method	2 (SK_MIC)
	Reserved	0
	Authentication Data	any
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	16391 (USE_TRANSPORT_MODE)
SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0

	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Table below

- SA Payload

SA Payload	Next Payload				44 (TSi)
	Critical				0
	Reserved				0
	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

Common Packet #4 : IKE_AUTH response for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	35 (IKE_AUTH)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	1
	Length	any
E Payload	Next Payload	36 (IDr)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
IDi Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	24
	ID Type	IPV6_ADDR
	Reserved	0
	Identification Data	TN1's Global Address on Link X
AUTH Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Auth Method	2 (SK_MIC)
	Reserved	0
	Authentication Data	any
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	16391(USE_TRANSPORT_MODE)
SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0

	Traffic Selectors	See Traffic Selector Table below

- SA Payload

SA Payload	Next Payload				44 (TSi)
	Critical				0
	Reserved				0
	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

Common Packet #5 : IKE_AUTH request for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	35 (IKE_AUTH)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	1
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	1
	Length	any
E Payload	Next Payload	35 (IDi)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
IDi Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	24
	ID Type	IPV6_ADDR
	Reserved	0
	Identification Data	TN1's Global Address on Link X
AUTH Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Auth Method	2 (SK_MIC)
	Reserved	0
	Authentication Data	any
SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40

	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSr Payload for End-Node to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)

		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- TSi Payload for SGW to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

- TSr Payload for SGW to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

Common Packet #6 : IKE_AUTH response for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	35 (IKE_AUTH)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	1
	Length	any
E Payload	Next Payload	36 (IDr)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
IDi Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	24
	ID Type	IPV6_ADDR
	Reserved	0
	Identification Data	TN1's Global Address on Link X
AUTH Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Auth Method	2 (SK_MIC)
	Reserved	0
	Authentication Data	any
SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	40

	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT’s Global Address on Link A
		Ending Address	NUT’s Global Address on Link A

- TSr Payload for End-Node to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)

		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSi Payload for SGW to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- TSr Payload for SGW to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

CREATE_CHILD_SA Messages for Generating CHILD_SA

Common Packet #7 : CREATE_CHILD_SA request for Generating CHILD_SA for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	16391(USE_TRANSPORT_MODE)
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload				Next Payload	44 (TSi)
				Critical	0
				Reserved	0
				Payload Length	40
	Proposal #1	SA Proposal		Next Payload	0 (last)
				Reserved	0
				Proposal Length	36
				Proposal #	1
				Proposal ID	3 (ESP)
				SPI Size	4
				# of Transforms	3
				SPI	any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

Common Packet #8 : CREATE_CHILD_SA response for Generating CHILD_SA for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	16391 (USE_TRANSPORT_MODE)
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0

	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT’s Global Address on Link A
		Ending Address	NUT’s Global Address on Link A

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1’s Global Address on Link X
		Ending Address	TN1’s Global Address on Link X

Common Packet #9 : CREATE_CHILD_SA request for Generating CHILD_SA for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload			44 (TSi)
	Critical			0
	Reserved			0
	Payload Length			40
	Proposal #1	SA Proposal	Next Payload	0 (last)
			Reserved	0
			Proposal Length	36
			Proposal #	1
			Proposal ID	3 (ESP)

			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

Common Packet #10 : CREATE_CHILD_SA response for Generating CHILD_SA for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	any
	Encrypted IKE Payloads	any
	Padding	any
	Pad Length	any
	Integrity Checksum Data	any
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload			44 (TSi)
	Critical			0
	Reserved			0
	Payload Length			40
	Proposal #1	SA Proposal	Next Payload	0 (last)
			Reserved	0
			Proposal Length	36
			Proposal #	1
			Proposal ID	3 (ESP)
			SPI Size	4

			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

CREATE_CHILD_SA Messages for Rekeying IKE_SA

Common Packet #11 : CREATE_CHILD_SA request for Rekeying IKE_SA

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	44
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any

- SA Payload

SA Payload	Next Payload				34 (KE)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		1 (IKE)
			SPI Size		0
			# of Transforms		4
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	2 (PRF)

				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

Common Packet #12 : CREATE_CHILD_SA response for Rekeying IKE_SA

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	44
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any

- SA Payload

SA Payload	Next Payload				34 (KE)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		1 (IKE)
			SPI Size		0
			# of Transforms		4
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	2 (PRF)
				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0

				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

CREATE_CHILD_SA Messages for Rekeying CHILD_SA

Common Packet #13 : CREATE_CHILD_SA request for Rekeying CHILD_SA for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	4
	Notify Message Type	16393 (REKEY_SA)
	SPI	any
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	16391 (USE_TRANSPORT_MODE)
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffice Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0

	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffice Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload				44 (TSi)
	Critical				0
	Reserved				0
	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

Common Packet #14 : CREATE_CHILD_SA response for Rekeying CHILD_SA for Transport Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum ofthe entire message
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	16391 (USE_TRANSPORT_MODE)
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffice Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffice Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0

	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for End-Node to End-Node test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- TSr Payload for End-Node to End-Node test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

Common Packet #15 : CREATE_CHILD_SA request for Rekeying CHILD_SA for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	4
	Notify Message Type	16393 (REKEY_SA)
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table below
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload	44 (TSi)
	Critical	0

	Reserved				0
	Payload Length				40
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		36
			Proposal #		1
			Proposal ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:ffff:ffff:ffff:ffff
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

Common Packet #16 : CREATE_CHILD_SA response for Rekeying CHILD_SA for Tunnel Mode

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
SA Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	40
	SA Proposals	See SA Payload Table
Ni, Nr Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	any
	Nonce Data	any
TSi Payload	Next Payload	45 (TSr)
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSi Payload Table below
TSr Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	48
	Number of TSs	1
	Reserved	0
	Traffic Selectors	See TSr Payload Table below

- SA Payload

SA Payload	Next Payload			44 (TSi)
	Critical			0
	Reserved			0
	Payload Length			40
	Proposal #1	SA Proposal	Next Payload	0 (last)
			Reserved	0
			Proposal Length	36
			Proposal #	1
			Proposal ID	3 (ESP)
			SPI Size	4

			# of Transforms		3
			SPI		any
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	1 (ENCR)
				Reserved	0
				Transform ID	3 (3DES)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (ESN)
				Reserved	0
				Transform ID	0 (No ESN)

- TSi Payload for SGW to SGW test cases

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- TSr Payload for SGW to SGW test cases

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

INFORMATIONAL Messages

Common Packet #17 : INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The value incremented the previous IKE message's Message ID by one. If this message is first one, this value is set to 0.
	Length	any
E Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message

Common Packet #18 : INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	The same value as corresponding request's IKE_SA Responder's SPI value
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The cryptographic checksum of the entire message

ICMPv6 Echo Requests

Common Packet #19 : ICMPv6 Echo Request for End-Node to End-Node test cases

IPv6 Header	Source Address	TN1's Global Address
	Destination Address	NUT's Global Address
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	58 (IPV6-ICMP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
ICMPv6 Header	Type	128
	Code	0
	Identifier	0
	Sequence Number	any
	Payload Data	0x0000000000000000

Common Packet #20 : ICMPv6 Echo Request for End-Node to SGW test cases

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TH1's Global Address
	Destination Address	NUT's Global Address on Link A
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Common Packet #21 : ICMPv6 Echo Request for SGW to SGW test cases

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
	Source Address	TH2's Global Address
IPv6 Header	Destination Address	TH1's Global Address
	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
ICMPv6 Header	Payload Data	0x0000000000000000

Common Packet #22 : ICMPv6 Echo Request for SGW to End-Node test cases

IPv6 Header	Source Address	TN1's Global Address

	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	58 (IPV6-ICMP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TN1's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

ICMPv6 Echo Replys

Common Packet #23 : ICMPv6 Echo Reply for End-Node to End-Node test cases

IPv6 Header	Source Address	TN1's Global Address
	Destination Address	NUT's Global Address
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	58 (IPV6-ICMP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
ICMPv6 Header	Type	129
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Common Packet #24 : ICMPv6 Echo Reply for End-Node to SGW test cases

IPv6 Header	Source Address	NUT's Global Address on Link A
	Destination Address	TN1's Global Address on Link X
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	NUT's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	129
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Common Packet #25 : ICMPv6 Echo Reply for SGW to SGW test cases

IPv6 Header	Source Address	TH1's Global Address
	Destination Address	TH2's Global Address
ICMPv6 Header	Type	129
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Common Packet #26 : ICMPv6 Echo Reply for SGW to End-Node test cases

IPv6 Header	Source Address	TH1's Global Address
	Destination Address	TN1's Global Address
ICMPv6 Header	Type	129
	Code	0
	Identifier	Any
	Sequence Number	Any
	Payload Data	0x0000000000000000

Section 1. End Node
Section 1.1. Initiator
Section 1.1.1. Endpoint-to-Endpoint Transport
Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.EN.I.1.1.1.1: Sending IKE_SA_INIT request

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT request using properly Header and Payloads format.

References:

- [RFC4306] - Section 1.2, 2.10, 3.1, 3.2, 3.3, 3.4 and 3.9
- [RFC 4718] - Sections 7.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |                |
    V                V
```

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Part B: SA Payload Format (BASIC)

3. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
4. Observe the messages transmitted on Link A.

Part C: KE Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.

Part D: Nonce Payload Format (BASIC)

7. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                      IKE_SA Initiator's SPI                   !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                      IKE_SA Responder's SPI                   !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1 Header format

- An IKE_SA Initiator's SPI field set to a 64-bits value chosen by the NUT. It MUST not be zero.
- An IKE_SA Responder's SPI field set to zero.
- A Next Payload field set to SA Payload (33).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to IKE_SA_INIT (34).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to zero.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 4: Judgment #1

```
                                 1                   2                   3
             0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
            ! Next     34    !0!       0     ! Length          44            !             |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
            !       0       !        0      ! Length          40            ! |           |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
            ! Number     1   ! Prot ID    1   ! SPI Size   0   ! Trans Cnt   4  ! |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | !       3       !        0      ! Length           8            ! |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | ! Type  1  (EN) !        0      ! Transform ID     3     (3DES) ! |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | !       3       !        0      ! Length           8            ! |           |SA Payload
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Proposal   |
          | ! Type  2  (PR) !        0      ! Transform ID     2     (SHA1) ! |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | !       3       !        0      ! Length           8            ! |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | ! Type  3  (IN) !        0      ! Transform ID     2     (SHA1) ! |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | !       0       !        0      ! Length           8            ! |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          | ! Type  4  (DH) !        0      ! Transform ID     2     (1024) ! |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 2 SA Payload contents

The NUT transmits an IKE_SA_INIT request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3 SA Payload format

- A Next Payload field is set to KE Payload (34).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 40 bytes for this proposal according to Common Configuration.
- A Proposal # field is set to 1 if this structure is the first proposal, otherwise set to 1 greater than the previous proposal.
- A Protocol ID field is set to IKE (1).
- A SPI Size field is set to zero.
- A # of Transforms field is set to 4.

A Transform field is set to following (There are 4 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for PRF_HMAC_SHA1.
- A Transform Type field is set to PRF (2).
- A RESERVED field is set to zero.
- A Transform ID set to PRF_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #4

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for 1024 MODP Group.
- A Transform Type field is set to D-H (4).
- A RESERVED field is set to zero.
- A Transform ID set to Group2 (2).

Part C

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT request including properly formatted KE Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          DH Group #           !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Key Exchange Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6 KE Payload format

- A Next Payload field is set to Nonce Payload (40).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 136 bytes for Group 2.
- A DH Group field is set to Group2 (2).
- A RESERVED field is set to zero.
- A Key Exchange Data field is set to Diffie-Hellman public value. The length of the Key Exchange Data field must be equal to 1024bit.

Part D

Step 8: Judgment #1

The NUT transmits an IKE_SA_INIT request including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7 Nonce Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Nonce Data field is set to random data generated by the transmitting entity. The size of the Nonce must between 16 and 256 octets.

Possible Problems:

- IKE_SA_INIT request has following packet format. It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(COOKIE)],
SA, KE, Ni,
[N(NAT_DETECTION_SOURCE_IP)+,
 N(NAT_DETECTION_DESTINATION_IP)],
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- Each of transforms can be located in the any order.

Test IKEv2.EN.I.1.1.1.2: Sending IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format.

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link A.

Part C: IDi Payload Format (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.

11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.

Part D: AUTH Payload Format (BASIC)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link A.

Part E: Notify Payload Format (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.

Part F: SA Payload Format (BASIC)

21. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link A.
23. TN1 responds with an IKE_SA_INIT response to the NUT.
24. Observe the messages transmitted on Link A.

Part G: TSi Payload Format (BASIC)

25. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. TN1 responds with an IKE_SA_INIT response to the NUT.
28. Observe the messages transmitted on Link A.

Part H: TSr Payload Format (BASIC)

29. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
30. Observe the messages transmitted on Link A.
31. TN1 responds with an IKE_SA_INIT response to the NUT.
32. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted IKE Header containing following values:

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8 Header format

- An IKE_SA Initiator's SPI field is set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field is set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to IKE_AUTH (35).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to 1.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted Encrypted Payload containing following values:

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9 Encrypted payload

- A Next Payload field is set to IDi Payload (35).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10 ID Payload format

- A Next Payload field is set to AUTH Payload (39).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field is set to ID_IPV6_ADDR (5).
- A RESERVED field is set to zero.
- An Identification Data field is set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11 AUTH Payload format

- A Next Payload field is set to Notify Payload (41).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field is set to Shared Key Message Integrity Code (2).
- A RESERVED field is set to zero.
- An Authentication Data field is set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12 Notify Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 8 bytes for USE_TRANSPORT_MODE.
- A Protocol ID field is set to undefined (0).
- A SPI Size field is set to zero.
- A Notify Message Type field is set to USE_TRANSPORT_MODE (16391)

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 24: Judgment #2

```
                                 1                   2                   3
             0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
            ! Next     44   !0!      0      ! Length          40            !            |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---         |
            !       0       !       0       ! Length          36            !  |         |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
            ! Number     1  ! Prot ID    3  ! SPI Size   4  ! Trans Cnt  3  !  |         |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
            ! SPI value                                                     !  |         |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
          | !       3       !       0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |SA Payload
          | ! Type  1  (EN) !       0       ! Transform ID     3      (3DES) !  | Proposal |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
          | !       3       !       0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
          | ! Type  3  (IN) !       0       ! Transform ID     2      (SHA1) !  |         |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
          | !       0       !       0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
          | ! Type  5  (ESN)!       0       ! Transform ID     0       (No)  !  |         |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 13 SA Payload contents

The NUT transmits an IKE_AUTH request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14 SA Payload format

- A Next Payload field is set to TSi Payload (44).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field is set to 1 if this structure is the first proposal, otherwise set to 1 greater than the previous proposal.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to 4.
- A # of Transforms field is set to 3.
- A SPI field is set to the sending entity's SPI (4 octets value)

Transform field is set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field is set to ESN (5).
- A RESERVED field is set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17 TSi Payload format

- A Next Payload field is set to TSr Payload (45).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to the number of actual traffic selectors.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to NUT address.
- A Ending Address field is set to greater thatn or equal to NUT address.

Part H

Step 30: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 32: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19 TSr Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to the number of actual traffic selectors.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to TN1 address.
- An Ending Address field is set to less than or equal to TN1 address.

Possible Problems:

- IKE_AUTH request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDi,
[CERT+],
[N(INITIAL_CONTACT)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
[IDr],
AUTH,
[CP(CFG_REQUEST)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA,
TSi,
TSr,
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.
- Each of transforms can be located in the any order.
- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

Test IKEv2.EN.I.1.1.1.3: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |<--------------| IPsec {Echo Request}
    |               | (Packet #3)
    |-------------->| IPsec {Echo Reply}
    |               | (Judgment #3)
    |               |
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.

7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Group 1.2. Use of Retransmission Timers

Test IKEv2.EN.I.1.1.2.1: Retransmissions of IKE_SA_INIT requests

Purpose:

To verify an IKEv2 device retransmits IKE_SA_INIT request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |                |
     |                * wait for the event of a timeout
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #2)
     |                |
     V                V
```

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 waits for the event of a timeout on NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request's Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.EN.I.1.1.2.2: Stop of retransmission of IKE_SA_INIT requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |                |
    |                * wait for the event of a timeout
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #2)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |                * wait for the event of a timeout
    |                |
    |-------X        | never send IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #3)
    |                |
    V                V
```

Packet #1	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 waits for the event of a timeout on NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response to the NUT.
6. TN1 waits for the event of a timeout on NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request’s Message ID value in IKE Header.

Step 7: Judgment #3
The NUT never retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.EN.I.1.1.2.3: Retransmissions of IKE_AUTH requests

Purpose:

To verify an IKEv2 device retransmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |                * wait for the event of a timeout
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 waits for the event of a timeout on NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.EN.I.1.1.2.4: Stop of retransmission of IKE_AUTH requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    |               * wait for the event of a timeout
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #3)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |               * wait for the event of a timeout
    |               |
    |-------X       | never send IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #4)
    |               |
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.

4. Observe the messages transmitted on Link A.
5. TN1 waits for the event of a timeout on NUT.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_AUTH response to the NUT.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Step 9: Judgment #4
The NUT never retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Group 1.3. State Synchronization and Connection Timeouts

Test IKEv2.EN.I.1.1.3.1: State Synchronization with ICMP messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives ICMP messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TR1              TN1
(End-Node)        (Router)        (End-Node)
     |                |                |
     |----------------+--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                |                | (Judgment #1)
     |<---------------+----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                |                | (Packet #1)
     |                |                |
     |----------------+--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                |                | (Judgment #2)
     |<---------------+----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                |                | (Packet #2)
     |                |                |
     |<---------------+----------------| IPsec {Echo Request}
     |                |                | (Packet #3)
     |----------------+--------------->| IPsec {Echo Reply}
     |                |                | (Judgment #3)
     |                |                |
     |<---------------|                | ICMPv6 Destination Unreachable (No route to destination)
     |                |                | (Packet #4)
     |                |                |
     |<---------------+----------------| IPsec {Echo Request}
     |                |                | (Packet #5)
     |----------------+--------------->| IPsec {Echo Reply}
     |                |                | (Judgment #4)
     |                |                |
     V                V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19

Packet #4	See below
Packet #5	See Common Packet #19

Packet #4: ICMPv6 Destination Unreachable

IPv6 Header	Source Address	TR1's Global Address on Link A
	Destination Address	NUT's Global Address on Link A
ICMPv6 Header	Type	1
	Code	0

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. After reception of an Echo Reply from NUT, TR1 transmits ICMP Destination Unreachable Message to the NUT and then TN1 transmits an Echo Request to the NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.3.2: State Synchronization with IKE messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives IKE messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #2)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #3)
     |                |
     |<---------------| cryptographically unprotected IKE message
     |                | (Packet #3)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #4)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #4)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See below
Packet #4	See Common Packet # 20

Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link A
	Destination Address	NUT's Global Address on Link X
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. TN1 transmits a cryptographically unprotected INFORMATIONAL request with Notify payload of type INVALID_ SPI to the NUT.
9. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #4
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None

Test IKEv2.EN.I.1.1.3.3: Close connections when repeated attempts fail

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  n each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |                 |
     |                 * wait for the event of a timeout
     |                 |
     |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #3)
     |                 |
     |                 * wait for the event of a timeout
     |                 |
     |-------X         | never send IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #3)
     |                 |
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 waits for the event of a timeout on the NUT.
6. Observe the messages transmitted on Link A.
7. Repeat Step 5 and Step 6 until the NUT’s last restransmission comes.

8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Step 8: Judgment #4
The NUT never retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.3.4: Close connections when receiving INITIAL_CONTACT

Purpose:

To verify an IKEv2 device closes connections when receiving INITIAL_CONTACT.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 7.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
 (End-Node)        (End-Node)
      |                |
      |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
      |                | (Judgment #1)
      |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
      |                | (Packet #1)
      |                |
      |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
      |                | (Judgment #2)
      |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
      |                | (Packet #2)
      |                |
      |<---------------| IPsec {Echo Request}
      |                | (Packet #3)
      |--------------->| IPsec {Echo Reply}
      |                | (Judgment #3)
      |                |
      |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
      |                | (Judgment #4)
      |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
      |                | (Packet #4)
      |                |
      |--------------->| IKE_AUTH request (HDR, SK {IDi, N(INITIAL_CONTACT), AUTH, N, SAi2, TSi, TSr})
      |                | (Judgment #5)
      |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
      |                | (Packet #5)
      |                |
      |<---------------| IPsec {Echo Request}
      |                | (Packet #6)
      |-------X        | IPsec {Echo Reply}
      |                | (Judgment #6)
      |                |
      |<---------------| IPsec {Echo Request}
      |                | (Packet #7)
      |--------------->| IPsec {Echo Reply}
      |                | (Judgment #7)
      |                |
      V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #2
Packet #5	See Common Packet #4
Packet #6	See Common Packet #19 This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 5.
Packet #7	See Common Packet #19 This packet is cryptographically protected by the CHILD_SA negotiated at Step 9 to Step 13.

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request. If rebooting NUT to start negotiation again is needed, it is possible to reboot NUT.
9. NUT transmits IKE_SA_INIT request to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.
13. After reception of IKE_AUTH Resquest from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
14. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithm.
15. Observe the messages transmitted on Link A.
16. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithm.
17. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #4
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #5
The NUT transmits IKE_AUTH request with a Notify payload of type INITIAL_CONTACT to the NUT. And the IKE_AUTH request includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 15: Judgment #6
The NUT never transmits an Echo Reply.

Step 17: Judgment #7
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithm.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.3.5: Sending Liveness check

Purpose:

To verify an IKEv2 device checks whether the other endpoint is alive.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |                * wait until receiving a liveness check
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Test IKEv2.EN.I.1.1.3.6: Sending Delete Payload for IKE_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when IKE_SA is deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 30 seconds.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |                * wait until expiring IKE_SA
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Possible Problems:

- At Step 7, NUT can transmit INFORMATIONAL request with a Delete Payload including 2 (ESP) as Protocol ID, 4 as SPI Size and SPI value to delete CHILD_SA before transmitting an INFORMATIONAL request to delete IKE_SA.

Test IKEv2.EN.I.1.1.3.7: Sending Delete Payload for CHILD_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |                * wait until expiring CHILD_SA
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT

6. TN1 waits until expiring CHILD_SA's lifetime and TN1 does not respond to an INFORMATIONAL reques with an INFORMATIONAL request for liveness check.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Possible Problems:

- None

Test IKEv2.EN.I.1.1.3.8: Sending Liveness check with unprotected messages

Purpose:

To verify an IKEv2 device handles cryptographically unprotected Messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  Configure the timer to consider that the peer is dead to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
   |                |
   |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Judgment #1)
   |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                | (Packet #1)
   |                |
   |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
   |                | (Judgment #2)
   |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
   |                | (Packet #2)
   |                |
   |<---------------| Echo Request
   |                | (Packet #3)
   |--------X       | Echo Reply
   |                | (Judgment #3)
   |                |
   |--------------->| INFORMATIONAL request (HDR, SK {})
   |                | (Judgment #4)
   |                |
   V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See below

Packet #3: Echo Request

IPv6 Header	Source Address	TN1's Global Address
	Destination Address	NUT's Global Address
ICMPv6 Header	Type	128

	Code	0
	Identifier	0
	Sequence Number	any
	Payload Data	0x0000000000000000

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 trasmits a cryptographically unprotected Echo Request to the NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT never responds with a cryptographically unprotected Echo Reply. The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- NUT may have the different trigger other than timer to send an INFORMATIONAL request for the liveness check. In that case, TN must be adjusted to support such a trigger.

Group 1.4. Version Numbers and Forward Compatibility

Test IKEv2.EN.I.1.1.4.1: Unrecognized payload types and Critical bit is not set

Purpose:

To verify an IKEv2 device ignores invalid payload types when the invalid type payload's critical bit is not set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                        ---
     |<---------------| IPsec {Echo Request}     |
     |                | (Packet #3)              |
     |--------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)            |
     |                |                        ---
    ...              ...
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Judgment #4)
     |<---------------| CREATE_CHILD_SA response (HDR, SK {P, N+, SA, Nr, TSi, TSr})
     |                | (Packet #4)
     |                |
     |<---------------| IPsec {Echo Request} (new CHILD_SA)
     |                | (Packet #5)
     |--------------->| IPsec {Echo Reply} (new CHILD_SA)
     |                | (Judgment #5)
     |                |
     V                V

P: Payload with an invalid payload type
N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See Common Pakcet #19

Packet #4: CREATE_CHILD_SA response

IPv6 Header	All fields are same as Common Packet #14 Payload	
UDP Header	All fields are same as Common Packet #14 Payload	
IKEv2 Header	All fields are same as Common Packet #14 Payload	
E payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #14	
Invalid Payload	Next Payoad	41 (N)
	Critical	0
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #14 Payload	
SA Payload	All fields are same as Common Packet #14 Payload	
Ni, Nr paylaod	All fields are same as Common Packet #14 Payload	
TSi Payload	All fields are same as Common Packet #14 Payload	
TSr Payload	All fields are same as Common Packet #14 Payload	

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 1 and the invalid payload's critical flag is not set.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
13. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 (BASIC)

14. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an IKE_SA_INIT response to the NUT.
17. Observe the messages transmitted on Link A.
18. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
19. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
20. Observe the messages transmitted on Link A.
21. Repeat Steps 19 and 20 until lifetime of SA is expired.
22. Observe the messages transmitted on Link A.
23. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 32 and the invalid payload's critical flag is not set.
24. Observe the messages transmitted on Link A.
25. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
26. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 (BASIC)

27. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
28. Observe the messages transmitted on Link A.
29. TN1 responds with an IKE_SA_INIT response to the NUT.
30. Observe the messages transmitted on Link A.
31. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
32. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
33. Observe the messages transmitted on Link A.
34. Repeat Steps 32 and 33 until lifetime of SA is expired.
35. Observe the messages transmitted on Link A.
36. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 49 and the invalid

payload's critical flag is not set.
37. Observe the messages transmitted on Link A.
38. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
39. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 (BASIC)

40. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
41. Observe the messages transmitted on Link A.
42. TN1 responds with an IKE_SA_INIT response to the NUT.
43. Observe the messages transmitted on Link A.
44. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
45. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
46. Observe the messages transmitted on Link A.
47. Repeat Steps 45 and 46 until lifetime of SA is expired.
48. Observe the messages transmitted on Link A.
49. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is not set.
50. Observe the messages transmitted on Link A.
51. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
52. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 13: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part B

Step 15: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 17: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 20 Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 24: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 26: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part C

Step 28: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 30: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 33 Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 37: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 39: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part D

Step 41: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 43: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 46 Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 50: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 52: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.4.2: Unrecognized payload types and Critical bit is set

Purpose:

To verify an IKEv2 device rejects the messages with invalid payload types when the invalid type payload's critical bit is set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                         ---
     |<---------------| IPsec {Echo Request}     |
     |                | (Packet #3)              |
     |--------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)            |
     |                |                         ---
    ...              ...
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Judgment #4)
     |<---------------| CREATE_CHILD_SA response (HDR, SK {P, N+, SA, Nr, TSi, TSr})
     |                | (Packet #4)
     |                |
     |<---------------| IPsec {Echo Request} (new CHILD_SA)
     |                | (Packet #5)
     |-------X        | IPsec {Echo Reply} (new CHILD_SA)
     |                | (Judgment #5)
     |                |
     V                V

P: Payload with an invalid payload type
N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See Common Pakcet #19

Packet #4: CREATE_CHILD_SA response

IPv6 Header	All fields are same as Common Packet #14 Payload	
UDP Header	All fields are same as Common Packet #14 Payload	
IKEv2 Header	All fields are same as Common Packet #14 Payload	
E payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #14	
Invalid Payload	Next Payoad	41 (N)
	Critical	1
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #14 Payload	
SA Payload	All fields are same as Common Packet #14 Payload	
Ni, Nr paylaod	All fields are same as Common Packet #14 Payload	
TSi Payload	All fields are same as Common Packet #14 Payload	
TSr Payload	All fields are same as Common Packet #14 Payload	

Part A: Invalid payload type 1 and Critical bit is set (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 1 and the invalid payload's critical flag is set.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
13. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 and Critical bit is set (BASIC)

14. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an IKE_SA_INIT response to the NUT.
17. Observe the messages transmitted on Link A.
18. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
19. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
20. Observe the messages transmitted on Link A.
21. Repeat Steps 19 and 20 until lifetime of SA is expired.
22. Observe the messages transmitted on Link A.
23. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 32 and the invalid payload's critical flag is set.
24. Observe the messages transmitted on Link A.
25. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
26. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 and Critical bit is set (BASIC)

27. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
28. Observe the messages transmitted on Link A.
29. TN1 responds with an IKE_SA_INIT response to the NUT.
30. Observe the messages transmitted on Link A.
31. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
32. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
33. Observe the messages transmitted on Link A.
34. Repeat Steps 32 and 33 until lifetime of SA is expired.
35. Observe the messages transmitted on Link A.
36. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 49 and the invalid

payload's critical flag is set.
37. Observe the messages transmitted on Link A.
38. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
39. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 and Critical bit is set (BASIC)

40. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
41. Observe the messages transmitted on Link A.
42. TN1 responds with an IKE_SA_INIT response to the NUT.
43. Observe the messages transmitted on Link A.
44. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
45. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
46. Observe the messages transmitted on Link A.
47. Repeat Steps 45 and 46 until lifetime of SA is expired.
48. Observe the messages transmitted on Link A.
49. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is set.
50. Observe the messages transmitted on Link A.
51. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to NUT.
52. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 13: Judgment #5
The NUT never transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part B

Step 15: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 17: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 20: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 24: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 26: Judgment #5
The NUT never transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part C

Step 28: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 30: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 33: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 37: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 39: Judgment #5
The NUT never transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Part D

Step 41: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 43: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 46: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 50: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 52: Judgment #5
The NUT never transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- None.

Group 1.5. Cookies

Test IKEv2.EN.I.1.1.5.1: Retrying IKE_SA_INIT request with a Notify payload of type COOKIE

Purpose:

To verify an IKEv2 device retries IKE_SA_INIT request using a Notify payload of type OOKIE.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT Response (HDR, N(COOKIE))
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
     |                 | (Judgment #2)
     |                 |
     V                 V
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header		All fields are same as Common Packet #2
UDP Header		All fields are same as Common Packet #2
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	0
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	34 (IKE_SA_INIT)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0

	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 4: Judgment #2

The NUT transmits an IKE_SA_INIT request including a Notify payload of type COOKIE ontaining following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21 Notify Payload Format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A SPI Size field is set to zero.
- A Notify Message Type field is set to COOKIE (16390).
- A Notification Data field is set to the TN1 supplied cookie data.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.5.2: Interaction of COOKIE and INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles a series of the Initial Exchanges using a Notify ayload of type COOKIE and type INVALID_KE_PAYLOAD.

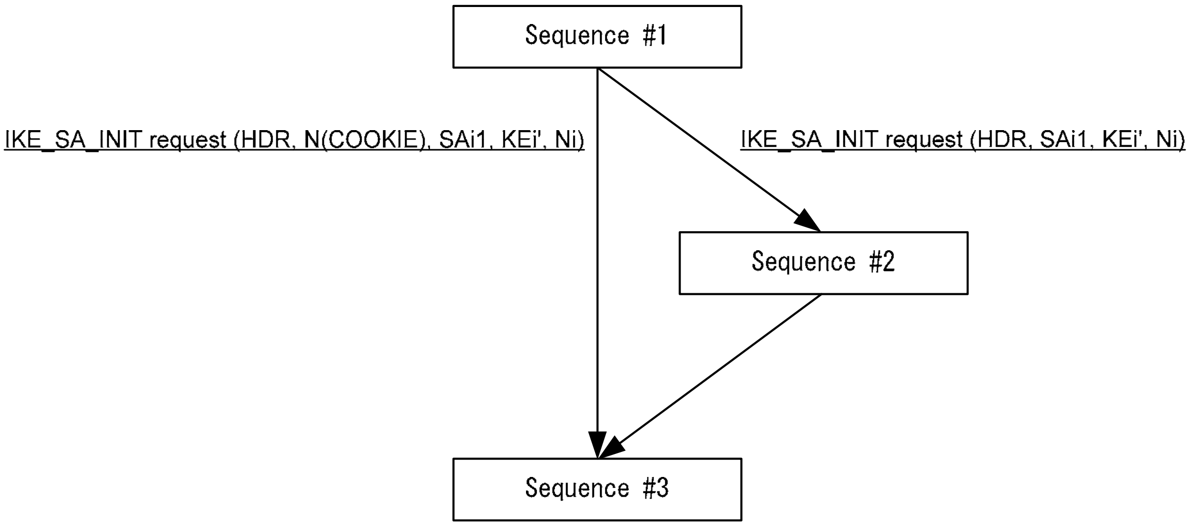
References:

- [RFC 4306] - Sections 2.6, 2.7 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:



```
Sequence #1:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Packet #1)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Judgment #2)
    |<---------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Packet #2)
    |                |
    |------*1------->| IKE_SA_INIT request (HDR, SAi1, KEi ' , Ni)
    |                |   or
    |------*2------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni)
    |                | (Judgment #3)
    |                |
    *1) If the NUT send IKE_SA_INIT request (HDR, SAi1, KEi ' , Ni), go to Sequence #2.
    *2) If the NUT send IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni), go to Sequence #3.
    Otherwise, this test is failed.

Sequence #2:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                | (Packet #3)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi ' , Ni)
    |                | (Judgment #4)
    |                |
    Go to Sequence #3.

Sequence #3:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #4)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #2

Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #2: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #3: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Different cookie value from Packet #1's cookie value.
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD to the NUT.
6. Observe the messages transmitted on Link A.

7. If the IKE_SA_INIT request from NUT includes a Notify payload of type COOKIE, TN1 responds with an IKE_SA_INIT response. The message has a different cookie value from the cookie value at Step3.
   A) Observe the messages transmitted on Link A.
   B) TN1 responds with an IKE_SA_INIT response.
8. If the IKE_SA_INIT request from NUT does not include a Notify payload of type COOKIE, TN1 responds with an IKE_SA_INIT response
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request. The message has a Notify payload of type COOKIE with the cookie data supplied by the responder as the first payload. All other payloads are unchanged.

Step 6: Judgment #3
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 5. All other payloads are unchanged.

Step 7A: Judgment #4
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 7. All other payloads are unchanged.

Step 9: Judgment #5
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.5.3: Interaction of COOKIE and INVALID_KE_PAYLOAD with unoptimized Responder

Purpose:

To verify an IKEv2 device properly handles a series of the Initial Exchanges using a Notify ayload of type COOKIE and type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.6, 2.7 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Packet #1)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Judgment #2)
    |<---------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Packet #2)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi ' , Ni)
    |                |   or
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni)
    |                | (Judgment #3)
    |                |
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                | (Packet #3)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi ' , Ni)
    |                | (Judgment #4)
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #4)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #2

Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #2: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #3: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Different cookie value from Packet #1's cookie value.
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD to the NUT.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response. The message has a different cookie value from the cookie value at Step3.
8. Observe the messages transmitted on Link A.
9. TN1 responds with an IKE_SA_INIT response.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request. The message has a Notify payload of type COOKIE with the cookie data supplied by the responder as the first payload. All other payloads are unchanged.

Step 6: Judgment #3
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 5.

Step 8: Judgment #4
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 7. All other payloads are unchanged.

Step 10: Judgment #5
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- None.

Group 1.6. Cryptographic Algorithm Negotiation

Test IKEv2.EN.I.1.1.6.1: Cryptographic Algorithm Negotiation for IKE_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part E, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	***ENCR_AES_CBC***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	***ENCR_AES_CTR***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	***PRF_AES128_CBC***	AUTH_HMAC_SHA1_96	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	***AUTH_AES_XCBC_96***	Group 2
Part E	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                  TN1
(End-Node)          (End-Node)
    |                    |
    |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                    | (Judgment #1)
    |<-------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                    | (Packet #1)
    |                    |
    |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                    | (Judgment #2)
    |                    |
    V                    V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link A.

Part C: PRF PRF_AES128_CBC (ADVANCED)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link A.

Part E: D-H Group Group 14 (ADVANCED)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 1.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CTR”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 5.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_AES128_CBC”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 9.

Part D

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_AES_XCBC_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 13.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as proposed algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 17.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.2: Cryptographic Algorithm Negotiation for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part F, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**Extended Sequence Numbers**
Part A	***ENCR_AES_CBC***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part B	***ENCR_AES_CTR***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part C	***ENCR_NULL***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part D	ENCR_3DES	***AUTH_AES_XCBC_96***	No Extended Sequence Numbers
Part E	ENCR_3DES	***NONE***	No Extended Sequence Numbers
Part F	ENCR_3DES	AUTH_HMAC_SHA1_96	***Extended Sequence Numbers***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
(End-Node)          (End-Node)
    |                   |
    |------------------>| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                   | (Judgment #1)
    |<------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                   | (Packet #1)
    |                   |
    |------------------>| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                   | (Judgment #2)
    |<------------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   | (Packet #2)
    |                   |
    |<------------------| IPsec {Echo Request}
    |                   | (Packet #3)
    |------------------>| IPsec {Echo Reply}
    |                   | (Judgment #3)
    |                   |
    V                   V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Packet #3	See Common Packet #19

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
9. Observe the messages transmitted on Link A.
10. TN1 responds with an IKE_SA_INIT response to the NUT.
11. Observe the messages transmitted on Link A.
12. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
13. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
14. Observe the messages transmitted on Link A.

Part C: Encryption Algorithm ENCR_NULL (ADVANCED)

15. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
16. Observe the messages transmitted on Link A.
17. TN1 responds with an IKE_SA_INIT response to the NUT.
18. Observe the messages transmitted on Link A.
19. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
20. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
21. Observe the messages transmitted on Link A.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

22. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
23. Observe the messages transmitted on Link A.
24. TN1 responds with an IKE_SA_INIT response to the NUT.
25. Observe the messages transmitted on Link A.
26. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
27. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
28. Observe the messages transmitted on Link A.

Part E: Integrity Algorithm NONE (ADVANCED)

29. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
30. Observe the messages transmitted on Link A.
31. TN1 responds with an IKE_SA_INIT response to the NUT.
32. Observe the messages transmitted on Link A.
33. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
34. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
35. Observe the messages transmitted on Link A.

Part F: Extended Sequence Numbers (ADVANCED)

36. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
37. Observe the messages transmitted on Link A.
38. TN1 responds with an IKE_SA_INIT response to the NUT.
39. Observe the messages transmitted on Link A.
40. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
41. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
42. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 9: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 11: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_AES_CTR”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 14: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 16: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 18: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_NULL”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 21: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part D

Step 23: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 25: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 28: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part E

Step 30: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 32: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “NONE” and “No xtended Sequence Numbers” as proposed algorithms.

Step 35: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part F

Step 37: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed lgorithms.

Step 30: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1” and “Extended Sequence Numbers” as proposed algorithms.

Step 42: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.3: Sending Multiple Transforms for IKE_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_SA_INIT request with multiple transforms for IKE_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_HMAC_SHA1** **PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 2** **Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                  TN1
(End-Node)           (End-Node)
     |                    |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                    | (Judgment #1)
     V                    V
```

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo-Random Functions (ADVANCED)

3. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.

6. Observe the messages transmitted on Link A.

Part D: Multiple D-H Groups (ADVANCED)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “PRF_AES128_CBC“AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “D-H group 2” and “D-H group 14” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.4: Sending Multiple Proposals for IKE_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple proposals for CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_SA_INIT exchanges Algorithms					
	Proposal	**Protocol ID**	**Encryption**	**PRF**	**Integrity**	**D-H Group**
Part A	Proposal #1	IKE	ENCR_ 3DES	PRF_ HMAC_SHA1	AUTH_ HMAC_SHA1_96	Group 2
	Proposal #2	**IKE**	**ENCR_ AES_CBC**	**PRF_ AES128_CBC**	**AUTH_ AES_XCBC_96**	**Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
(End-Node)          (End-Node)
    |                    |
    |--------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                    |  (Judgment #1)
    |                    |
    V                    V

N: USE_TRANSPORT_MODE
```

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” in SA Proposal #1 (ESP) and “ENCR_AES_CBC”, “PRF_AES128_CBC”, “AUTH_AES_XCBC_96” and “D-H group 14” in SA Proposal #2 (ESP) as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.5: Sending Multiple Transforms for CHILD_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple transforms for CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
(End-Node)        (End-Node)
     |                  |
     |----------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Judgment #1)
     |<-----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                  | (Packet #1)
     |                  |
     |----------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                  | (Judgment #2)
     |                  |
     V                  V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
2. Observe the messages transmitted on Link A.
3. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.

4. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
6. Observe the messages transmitted on Link A.
7. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.
8. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (ADVANCED)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
10. Observe the messages transmitted on Link A.
11. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “No Extended Sequence Numbers” and “Extended Sequence Number” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.6: Sending Multiple Proposals for CHILD_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple proposals for CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_AUTH exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	**ESP**	**ENCR_AES_CBC**	**AUTH_AES_XCBC_96**	**ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                   TN1
(End-Node)           (End-Node)
    |                     |
    |-------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                     |  (Judgment #1)
    |<--------------|  IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                     |  (Packet #1)
    |                     |
    |-------------->|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                     |  (Judgment #2)
    |                     |
    V                     V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” in SA Proposal #1 (ESP) and then “ENCR_AES_CBC”, “AUTH_AES_XCBC_96” and “Extended Sequence Numbers” in SA Proposal #2 (ESP) as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.7: Receipt of INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                        ---
     |<--------------| IPsec {Echo Request}     |
     |                | (Packet #3)              |
     |-------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)            |
     |                |                        ---
    ...              ...
     |                |
     |-------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
     |                | (Judgment #4)
     |<--------------| CREATE_CHILD_SA response (HDR, SK {N(INVALID_KE_PAYLOAD)})
     |                | (Packet #4)
     |                |
     |-------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi ' , TSi, TSr})
     |                | (Judgment #5)
     |                |
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below

Packet #4: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #14
UDP Header		Same as Common Packet #14
IKEv2 Header		Same as Common Packet #14
E Payload		Same as Common Packet #14
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed

algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.8: Receipt of NO_PROPOSAL_CHOSEN

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT response with a Notify payload of type NO_PROPOSAL_CHOSEN.

References:

- [RFC 4306] - Sections 3.10.1
- [RFC 4718] - Sections 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
 (End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                 | (Packet #2)
     |                 |
    ...               ...
     |                 |                          ---
     |<----------------| IPsec {Echo Request}      |
     |                 | (Packet #3)               |
     |---------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
     |                 | (Judgment #3)             |
     |                 |                          ---
    ...               ...
     |                 |
     |---------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
     |                 | (Judgment #4)
     |<----------------| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
     |                 | (Packet #4)
     |                 |
     |---------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi ', TSi, TSr})
     |                 | (Judgment #5)
     |                 |
     V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below

Packet #4: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #14
UDP Header		Same as Common Packet #14
IKEv2 Header		Same as Common Packet #14
E Payload		Same as Common Packet #14
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	NO_PROPOSAL_CHOSEN (14)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type NO_PROPOSAL_CHOSEN to the NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed

algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. The new CREATE_CHILD_SA request is not a retransmitted request.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.9: Response with inconsistent SA proposal for IKE_SA

Purpose:

To verify an IKEv2 device properly handles a response with a SA payload which is inconsistent with one of its proposals.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)        (End-Node)
     |                 |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |-------X         | IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |                 |
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header	Same as the Common Packet #2
UDP Header	Same as the Common Packet #2
IKEv2 Header	Same as the Common Packet #2
SA Payload	See below
KEi Payload	Same as the Common Packet #2
Ni Payload	Same as the Common Packet #2

SA Payload	Next Payload			34 (KE)
	Critical			0
	Reserved			0
	Payload Length			44
	Proposal #1	SA Proposal	Next Payload	0 (last)
			Reserved	0
			Proposal Length	40
			Proposal #	1
			Protocol ID	1 (IKE)

			SPI Size		0
			# of Transforms		4
			SA Transform		See below
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	2 (PRF)
				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		12
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT. But the response includes a SA payload which has a different Transform ID from the proposed one.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT never transmits an IKE_AUTH request.

Possible Problems:

- Step 4
  The NUT may transmit or retransmit an IKE_SA_INIT request.

Test IKEv2.EN.I.1.1.6.10: Response with inconsistent proposal for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a response with a SA payload which is inconsistent with one of its proposals.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #3)
     |-------X        | IPsec {Echo Reply}
     |                | (Judgment #3)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #19

Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #4
UDP Header	Same as the Common Packet #4
IKEv2 Header	Same as the Common Packet #4
E Payload	Same as the Common Packet #4
IDr Payload	Same as the Common Packet #4
AUTH Payload	Same as the Common Packet #4

N Payload	Same as the Common Packet #4
SA Payload	See below
TSi Payload	Same as the Common Packet #4
TSr Payload	Same as the Common Packet #4

SA Payload	Next Payload				44 (TSi)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SA Transform		See below
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (Extended Sequence Number)
				Reserved	0
				Transform ID	0 (No Extended Sequence Number)

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		12
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_AUTH response to the NUT. But the response includes a SA payload which has a different Transform ID from the proposed one.
6. TN1 transmits an Echo Request with IPsec ESP using ENCR_AES_CBC and AUTH_HMAC_SHA1_96.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT never transmits an Echo Reply with IPsec ESP using ENCR_AES_CBC and AUTH_HMAC_SHA1_96.

Possible Problems:

- Step 7
  The NUT may transmit or retransmit an IKE_AUTH request. And the NUT may notify INVALID_SPI.

Test IKEv2.EN.I.1.1.6.11: Receipt of INVALID_KE_PAYLOAD in Initial Exchange

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT Response with a Notify payload of type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
 (End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi', Ni)
     |                 | (Judgment #2)
     |                 |
     V                 V
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header		Same as Common Packet #2
UDP Header		Same as Common Packet #2
IKEv2 Header		Same as Common Packet #2
	IKE_SA Responder's SPI	See each Part
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)

Part A: IKE_SA Responder's SPI is zero (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

3. TN1 responds with an IKE_SA_INIT Response including a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT. The message's IKE_SA Responder's SPI is set to zero.
4. Observe the messages transmitted on Link A.

Part B: IKE_SA Responder's SPI is not zero (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT Response including a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT. The message's IKE_SA Responder's SPI is set to one.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Part B

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.6.12: Creating an IKE_SA without a CHILD_SA

Purpose:

To verify an IKEv2 device can handles a failure of creating a CHILD_SA during the IKE_AUTH exchange.

References:

- [RFC 4718] - Sections 4.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)       (End-Node)
   |                |
   |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Judgment #1)
   |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                | (Packet #1)
   |                |
   |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
   |                | (Judgment #2)
   |<--------------| IKE_AUTH Response (HDR, N(NO_PROPOSAL_CHOSEN))
   |                | (Packet #2)
   |                |
   |<--------------| INFORMATIONAL request (HDR, SK {})
   |                | (Packet #3)
   |-------------->| INFORMATIONAL response (HDR, SK {})
   |                | (Judgment #3)
   |                |
   V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #17

Packet #4: IKE_AUTH response

IPv6 Header		Same as Common Packet #4
UDP Header		Same as Common Packet #4
IKEv2 Header		Same as Common Packet #4
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8

	Protocol ID	0
	SPI Size	0
	Notify Message Type	NO_PROPOSAL_CHOSEN (14)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response with a Notify payload of type NO_PROPOSAL_CHOSEN to the NUT.
6. TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Group 1.7. Traffic Selector Negotiation

Test IKEv2.EN.I.1.1.7.1: Narrowing the range of members of the set of traffic selectors

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)        (End-Node)
    |                 |
    |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
    |<----------------| IPsec {TCP SYN}
    |                 | (Packet #3)
    |---------------->| IPsec {TCP RST}
    |                 | (Judgment #3)
    |                 |
    |<----------------| IPsec {ICMPv6 Echo Request}
    |                 | (Packet #4)
    |-------X         | IPsec {ICMPv6 Echo Reply}
    |                 | (Judgment #4)
    |                 |
    V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below

Packet #3	See below
Packet #4	See Common Packet #19

Packet #2: IKE_AUTH response

TSi Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (tcp)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload			
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (tcp)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

Packet #3: TCP-SYN

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The cryptographic checksum of the entire message
TCP Header	Source Port	500
	Destination Port	500
	Flags	SYN (0x02)

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port on NUT.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT never transmit an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.8. Error Handling

Test IKEv2.EN.I.1.1.8.1: INVALID_IKE_SPI

Purpose:

To verify an IKEv2 device properly handles an unrecognized destination SPI.

References:

- [RFC 4306] - Sections 2.21 and 3.10.1
- [RFC 4718] - Sections 7.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
   ...             ...
    |               |                         ---
    |<--------------| IPsec {Echo Request}     |
    |               | (Packet #3)              |
    |-------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
    |               | (Judgment #3)            |
    |               |                         ---
   ...             ...
    |               |
    |-------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
    |               | (Judgment #4)
    |<--------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, KEr, TSi, TSr})
    |               | (Packet #4)
    |               |
    |-------X       | INFORMATIONAL request (HDR, SK {N(INVALID_IKE_SPI)})
    |               | (Judgment #5)
    |               |
    V               V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below

Part A

Packet #4: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #14
UDP Header		Same as Common Packet #14
IKEv2 Header	IKE_SA Initator's SPI	The IKE_SA Initiator's SPI value used by this IKE message plus 1
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
		Other field are same as Common Packet #14
E Payload		Same as Common Packet #14
N Payload		Same as Common Packet #14
SA Payload		Same as Common Packet #14
Ni, Nr Payload		Same as Common Packet #14
TSi Payload		Same as Common Packet #14
TSr Payload		Same as Common Packet #14

Part B

Packet #4: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #14
UDP Header		Same as Common Packet #14
IKEv2 Header	IKE_SA Initator's SPI	The IKE_SA Initiator's SPI value

		used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message plus 1
	Other field are same as Common Packet #14	
E Payload	Same as Common Packet #14	
N Payload	Same as Common Packet #14	
SA Payload	Same as Common Packet #14	
Ni, Nr Payload	Same as Common Packet #14	
TSi Payload	Same as Common Packet #14	
TSr Payload	Same as Common Packet #14	

Part A: Different IKE_SA Initiator's SPI (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which has an invalid value as IKE_SA Initiator's SPI to the NUT.
11. Observe the messages transmitted on Link A.

Part B: Different IKE_SA Responser's SPI (BASIC)

12. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link A.
16. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
17. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
18. Observe the messages transmitted on Link A.
19. Repeat Steps 6 and 7 until lifetime of SA is expired.
20. Observe the messages transmitted on Link A.
21. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which has an invalid value as IKE_SA Responder's SPI to the NUT.
22. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
NUT does not transmit any packets or may transmit INFORMATIONAL request with a Notify payload of typeINVALID_IKE_SPI.

Part B

Step 13: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 15: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 18: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 20: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 22: Judgment #5
NUT does not transmit any packets or may transmit INFORMATIONAL request with a Notify payload of typeINVALID_IKE_SPI.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.8.2: INVALID_SELECTORS

Purpose:

To verify an IKEv2 device properly handles an ESP or AH packet whose selectors do not match those of the CHILD_SA.

References:

- [RFC 4306] - Sections 3.10.1
- [RFC 4307] - Sections 7.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	TSi	TSr
IP Protocol ID	***IPv6-ICMP***	***IPv6-ICMP***
Start Port	0	0
End Port	65535	65535
Starting Address	TH1	NUT
Ending Address	TH1	NUT

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)       (End-Node)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |<--------------| IPsec {TCP SYN}
    |                | (Packet #3)
    |-------X        | No Packets
    |                | (Judgment #3)
    |                |   or
    |-------------->| INFORMATIONAL request (HDR, SK {N(INVALID_SELECTORS)})
    |                | (Judgment #3)
    |                |
    |<--------------| IPsec {Echo Request}
    |                | (Packet #4)
    |-------------->| IPsec {Echo Reply}
    |                | (Judgment #4)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See below
Packet #4	See Common Packet #19

Packet #3: TCP-SYN

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The cryptographic checksum of the entire message
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH

response to the NUT
6. TN1 transmits a TCP-SYN packet t with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT does not transmit any packets or transmits an INFORMATIONAL request with a Notify of type INVALID_SELECTORS.

Step 9: Judgment #4
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- Notification Type depends on the implementation at Step 7.
- If the NUT uses TCP port 30000 for other applications, the TN1 transmits TCP-SYN packets to other closed TCP port on the NUT.

Group 1.9 Identification Payload

Test IKEv2.EN.I.1.1.9.1: Sending IDi Payload

Purpose:

To verify an IKEv2 device transmits IDi payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDi payload which NUT sends to TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)       (End-Node)
     |                 |
     |--------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 |  (Judgment #1)
     |<---------------|  IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 |  (Packet #1)
     |                 |
     |--------------->|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 |  (Judgment #2)
     |                 |
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: ID_IPV4_ADDR (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.

4. Observe the messages transmitted on Link A.

Part B: ID_FQDN (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link A.

Part C: ID_RFC822_ADDR (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.9.2: Receiving IDr Payload

Purpose:

To verify an IKEv2 device receives IDr payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDr payload which NUT receives from TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                  TN1
(End-Node)          (End-Node)
   |                    |
   |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                    | (Judgment #1)
   |<-------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                    | (Packet #1)
   |                    |
   |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
   |                    | (Judgment #2)
   |<-------------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
   |                    | (Packet #2)
   |                    |
   |<-------------------| IPsec {Echo Request}
   |                    | (Packet #3)
   |------------------->| IPsec {Echo Reply}
   |                    | (Judgment #3)
   |                    |
   V                    V

N: USE_TRNASPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #19

Packet #2: IKE_AUTH response

IPv6 Header	Same as Common Packet #4

UDP Header		Same as Common Packet #4
IKEv2 Header		Same as Common Packet #4
E Payload		Same as Common Packet #4
IDi Payload	ID Type	Depends on each test
	Identification Data	Depends on each test
		Same as Common Packet #4
AUTH Payload		Same as Common Packet #4
N Payload		Same as Common Packet #4
SA Payload		Same as Common Packet #4
TSi Payload		Same as Common Packet #4
TSr Payload		Same as Common Packet #4

Part A: ID_IPV4_ADDR (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDr payload as describe above to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Part B: ID_FQDN (BASIC)

8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
9. Observe the messages transmitted on Link A.
10. TN1 responds with an IKE_SA_INIT response to the NUT.
11. Observe the messages transmitted on Link A.
12. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDi payload as describe above to the NUT
13. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
14. Observe the messages transmitted on Link A.

Part C: ID_RFC822_ADDR (BASIC)

15. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
16. Observe the messages transmitted on Link A.
17. TN1 responds with an IKE_SA_INIT response to the NUT.
18. Observe the messages transmitted on Link A.
19. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDi payload as describe above to the NUT
20. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
21. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 7: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 9: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 11: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 14: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 16: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 18: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 21: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.10 Authentication of the IKE_SA

Test IKEv2.EN.I.1.1.10.1: Sending CERT Payload

Purpose:

To verify an IKEv2 device handles CERTREQ payload and transmits CERT payload properly.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Remote	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                 |
    |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr, CERTREQ)
    |                 | (Packet #1)
    |                 |
    |---------------->| IKE_AUTH request (HDR, SK {IDi, CERT, AUTH, N, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |                 |
    V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header	Same as the Common Packet #2	
UDP Header	Same as the Common Packet #2	
IKEv2 Header	Same as the Common Packet #2	
SA Payload	Same as the Common Packet #2	
KE Payload	Same as the Common Packet #2	
Nr Payload	Next Payload	38 (CERTREQ)
	Other fields are same as the Common Packet #2	

CERTREQ Payload	See below

CERTREQ Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Authority	any

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response with a CERTREQ payload to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request with a CERT payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding and the NUT’s certificate as Certificate Data.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.10.2: Sending CERTREQ Payload

Purpose:

To verify an IKEv2 device transmits CERTREQ payload and handles CERT payload properly.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.10.3: RSA Digital Signature

Purpose:

To verify an IKEv2 device authenticates the corresponding node by RSA Digital Signature.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, CERT, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #3)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #19

Packet #2: IKE_AUTH response

IPv6 Header	Same as Common Packet #4
UDP Header	Same as Common Packet #4

IKEv2 Header	Same as Common Packet #4	
E Payload	Same as Common Packet #4	
IDr Payload	Next Payload	37 (CERT)
	Other fields are same as the Common Packet #4	
CERT Payload	See below	
AUTH Payload	Same as Common Packet #4	
N Payload	Same as Common Packet #4	
SA Payload	Same as Common Packet #4	
TSi Payload	Same as Common Packet #4	
TSr Payload	Same as Common Packet #4	

CERT Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Data	TN1's X.509 Certificate

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDr payload as describe above to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.10.4: HEX string PSK

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 2.15

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Key Value
Remote	0xabadcafeabadcafeabadcafeabadcafe (128 bit binary string)

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)      (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Group 1.11. Invalid values

Test IKEv2.EN.I.1.1.11.1: Non zero RESERVED fields in IKE_SA_INIT response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                  TN1
(End-Node)           (End-Node)
     |                    |
     |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                    | (Judgment #1)
     |<-------------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                    | (Packet #1)
     |                    |
     |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                    | (Judgment #2)
     |                    |
     V                    V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2 All RESERVED fields are set to one.

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response whose RESERVED fields are set to one to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.11.2: Non zero RESERVED fields in IKE_AUTH response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #3)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4 All RESERVED fields are set to one.
Packet #3	See Common Packet #19

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response whose RESERVED fields are set to one to the NUT

6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.11.3: Version bit is set

Purpose:

To verify an IKEv2 device ignores the content of Version bit in IKE messages.

References:

- [RFC 4306] - Sections 3.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)       (End-Node)
    |                 |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |                 |
    V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2 Version bit is set to one.

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response whose Version bit is set to one to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.11.4: Unrecognized Notify Message Type of Error

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type intended for reporting error.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)       (End-Node)
     |                 |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, N+)
     |                 | (Packet #1)
     |                 |
     |-------X         | IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |                 |
     V                 V

N: USE_TRANSPORT_MODE
N+: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #2	
UDP Header	All fields are same as Common Packet #2	
IKEv2 Header	All fields are same as Common Packet #2	
SA Payload	All fields are same as Common Packet #2	
KE Payload	All fields are same as Common Packet #2	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #2	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	16383

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response with a Notify payload of unrecognized Notify Message Type value.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT never transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.1.11.5: Unrecognized Notify Message Type of Status

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type intended for reporting status.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                  TN1
(End-Node)           (End-Node)
     |                    |
     |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                    | (Judgment #1)
     |<-------------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, N+)
     |                    | (Packet #1)
     |                    |
     |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                    | (Judgment #2)
     |                    |
     V                    V

N: USE_TRANSPORT_MODE
N+: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #2	
UDP Header	All fields are same as Common Packet #2	
IKEv2 Header	All fields are same as Common Packet #2	
SA Payload	All fields are same as Common Packet #2	
KE Payload	All fields are same as Common Packet #2	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #2	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	65535

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response with a Notify payload of unrecognized Notify Message Type value.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Group 2. The CREATE_CHILD_SA Exchange

Group 2.1. Header and Payload Formats

Test IKEv2.EN.I.1.2.1.1: Sending CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device transmits CREATE_CHILD_SA request using properly Header and Payloads format.

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |---------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 |  (Judgment #1)
    |<----------------|  IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 |  (Packet #1)
    |                 |
    |---------------->|  IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 |  (Judgment #2)
    |<----------------|  IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 |  (Packet #2)
    |                 |
   ...               ...
    |                 |                          ---
    |<----------------|  IPsec {Echo Request}      |
    |                 |  (Packet #3)               |
    |---------------->|  IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                 |  (Judgment #3)             |
    |                 |                          ---
   ...               ...
    |                 |
    |---------------->|  CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                 |  (Judgment #4)
    |                 |
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packets #4
Packet #3	See Common Packets #19

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired for 30 seconds.
9. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

10. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link A.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
15. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
16. Observe the messages transmitted on Link A.
17. Repeat Steps 15 and 16 until lifetime of SA is expired for 30 seconds.
18. Observe the messages transmitted on Link A.

Part C: Notify Payload (REKEY_SA) Format (BASIC)

19. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. TN1 responds with an IKE_SA_INIT response to the NUT.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
24. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
25. Observe the messages transmitted on Link A.
26. Repeat Steps 24 and 25 until lifetime of SA is expired for 30 seconds.
27. Observe the messages transmitted on Link A.

Part D: Notify Payload (USE_TRANSPORT_MODE) Format (BASIC)

28. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
29. Observe the messages transmitted on Link A.
30. TN1 responds with an IKE_SA_INIT response to the NUT.
31. Observe the messages transmitted on Link A.
32. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
33. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
34. Observe the messages transmitted on Link A.
35. Repeat Steps 33 and 34 until lifetime of SA is expired for 30 seconds.
36. Observe the messages transmitted on Link A.

Part E: SA Payload Format (BASIC)

37. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
38. Observe the messages transmitted on Link A.
39. TN1 responds with an IKE_SA_INIT response to the NUT.
40. Observe the messages transmitted on Link A.
41. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
42. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
43. Observe the messages transmitted on Link A.
44. Repeat Steps 42 and 43 until lifetime of SA is expired for 30 seconds.
45. Observe the messages transmitted on Link A.

Part F: Nonce Payload Format (BASIC)

46. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
47. Observe the messages transmitted on Link A.
48. TN1 responds with an IKE_SA_INIT response to the NUT.
49. Observe the messages transmitted on Link A.
50. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
51. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
52. Observe the messages transmitted on Link A.
53. Repeat Steps 51 and 52 until lifetime of SA is expired for 30 seconds.
54. Observe the messages transmitted on Link A.

Part G: TSi Payload Format (BASIC)

55. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
56. Observe the messages transmitted on Link A.
57. TN1 responds with an IKE_SA_INIT response to the NUT.
58. Observe the messages transmitted on Link A.

59. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
60. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
61. Observe the messages transmitted on Link A.
62. Repeat Steps 60 and 61 until lifetime of SA is expired for 30 seconds.
63. Observe the messages transmitted on Link A.

Part H: TSr Payload Format (BASIC)

64. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
65. Observe the messages transmitted on Link A.
66. TN1 responds with an IKE_SA_INIT response to the NUT.
67. Observe the messages transmitted on Link A.
68. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
69. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
70. Observe the messages transmitted on Link A.
71. Repeat Steps 69 and 70 until lifetime of SA is expired for 30 seconds.
72. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted IKE Header containing following values:

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.
- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to CREATE_CHILD_SA (36).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to the value incremented the previous IKE message’s Message ID by one.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 11: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 16: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 18: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23 Encrypted payload

- A Next Payload field is set to N Payload (41).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 25: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 27: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24 Notify Payload format

- A Next Payload field is set to N Payload (41).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 12 bytes for this REKEY_SA.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to the size of CHILD_SA Inbound SPI value to be rekeyed. It is 4 bytes for ESP.
- A Notify Message Type field is set to REKEY_SA (16393).
- A Security Parameter Index field is set to SPI value to be rekeyed.
- A Notification Data field is empty.

Part D

Step 29: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 31: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 34: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 36: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 25 Notify Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 8 bytes for USE_TRANSPORT_MODE.
- A Protocol ID field is set to undefined (0).
- A SPI Size field is set to zero.
- A Notify Message Type field is set to USE_TRANSPORT_MODE (16391)

Part E

Step 38: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 40: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 43: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 45: Judgment #4

```
                         1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
    ! Next     44    !0!       0       ! Length           40           !              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---            |
    !       0        !         0       ! Length           36           !  |           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
    ! Number     1   ! Prot ID    3    ! SPI Size   4    ! Trans Cnt  3  !  |           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
    ! SPI value                                                        !  |           |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3        !         0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |SA Payload
          | ! Type  1  (EN) !         0       ! Transform ID     3      (3DES) !  | Proposal  |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3        !         0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  3  (IN) !         0       ! Transform ID     2      (SHA1) !  |           |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       0        !         0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  5  (ESN)!         0       ! Transform ID     0      (No)   !  |           |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 26 SA Payload contents

The NUT transmits a CREATE_CHILD_SA request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 27 SA Payload format

- A Next Payload field is set to Ni Payload (40).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 28 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field is set to 1 if this structure is the first proposal, otherwise set to 1 greater thatn the previous proposal.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to 4.
- A # of Transforms field is set to 3.
- A SPI field is set to the sending entity's SPI (4 octets value)

Transform field is set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 29 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.

- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field is set to ESN (5).
- A RESERVED field is set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 47: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 49: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 52: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 54: Judgment #4

The NUT transmits a CREATE_CHILD_SA request including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30 Nonce Payload format

- A Next Payload field is set to TSi Payload (44).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Nonce Data field is set to random data generated by the transmitting entity.
- The size of the Nonce must between 16 and 256 octets.

Part G

Step 56: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 58: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 61: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 63: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 31 TSi Payload format

- A Next Payload field is set to TSr Payload (45).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to the number of actual traffic selectors.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 32 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to NUT address.
- A Ending Address field is set to greater thatn or equal to NUT address.

Part H

Step 65: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 67: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 70: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 72: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 33 TSr Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to 1.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 34 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to TN1 address.
- An Ending Address field is set to less than or equal to TN1 address.

Possible Problems:

- The implementation may use different SA lifetimes by the implementation policy. In that case, the tester must change the expiration time to wait CREATE_CHILD_SA request.

- CREATE_CHILD_SA request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(REKEY_SA)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, Ni, [KEi], TSi, TSr
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- Each of transforms can be located in the any order.

- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

Group 2.2. Use of Retransmission Timers

Test IKEv2.EN.I.1.2.2.1: Retransmissions of CREATE_CHILD_SA requests

Purpose:

To verify an IKEv2 device retransmits CREATE_CHILD_SA request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                       ---
    |<---------------| IPsec {Echo Request}     |
    |                | (Packet #3)              |
    |--------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)            |
    |                |                       ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |                |
    |                * wait for the event of a timeout
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 waits for the event of a timeout on NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 11: Judgment #5
The NUT retransmits a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different lifetime of SA.
- Each NUT has the different retransmission timers.

Test IKEv2.EN.I.1.2.2.2: Stop of retransmission of CREATE_CHILD_SA requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
 (End-Node)      (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                             ---
     |<---------------| IPsec {Echo Request}         |
     |                | (Packet #3)                  |
     |--------------->| IPsec {Echo Reply}           | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)                |
     |                |                             ---
    ...              ...
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Judgment #4)
     |                |
     |                * wait for the event of a timeout
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Judgment #5)
     |<---------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
     |                | (Packet #4)
     |                |
     |                * wait for the event of a timeout
     |                |
     |-------X        | never send CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Judgment #6)
     |                |
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 waits for the event of a timeout on NUT.
11. Observe the messages transmitted on Link A.

12. TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. TN1 waits for the event of a timeout on NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 11: Judgment #5
The NUT retransmits a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Step 14: Judgment #6
The NUT stops the retransmissions of a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different lifetime of SA.
- Each NUT has the different retransmission timers.

Group 2.3. Rekeying CHILD_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.EN.I.1.2.3.1: Close the replaced CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Packet #4)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #5)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response with a Delete payload to the NUT.
13. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Step 14: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.2: Use of the new CHILD_SA

Purpose:

To verify an IKEv2 device properly rekeys CHILD_SA

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
   ...               ...
    |                 |                              ---
    |<---------------| IPsec {Echo Request}          |
    |                 | (Packet #3)                   |
    |--------------->| IPsec {Echo Reply}            | repeat Echo exchange until lifetime of SA is expired
    |                 | (Judgment #3)                 |
    |                 |                              ---
   ...               ...
    |                 |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                 | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                 | (Packet #4)
    |                 |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                 | (Judgment #5)
    |<---------------| INFORMATIONAL response (HDR, SK {D})
    |                 | (Packet #5)
    |                 |
    |<---------------| IPsec {Echo Request}
    |                 | (Packet #6)
    |--------------->| IPsec {Echo Reply}
    |                 | (Judgment #6)
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14
Packet #5	See below
Packet #6	See Common Packet #19 This packet is cryptographically protected by the new CHILD_SA negotiated at Step 10.

Packet #5: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0

	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response with a Delete payload to the NUT.
13. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Step 14: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.3: Lifetime of CHILD_SA expires

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of CHILD_SAs.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |<--------------| IPsec {Echo Request}
    |               | (Packet #3)
    |-------------->| IPsec {Echo Reply}
    |               | (Judgment #3)
    |               |
    |               * wait for the event of a timeout of CHILD_SA
    |               |
    |<--------------| IPsec {Echo Request}
    |               | (Packet #4)
    |--------X      | IPsec {Echo Reply}
    |               | (Judgment #4)
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #19

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. TN1 waits for the event of a timeout on the NUT.
9. After timeout of CHILD_SA on the NUT, TN1 transmits an Echo Request with IPsec ESP which has expired to the NUT.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #4
The NUT does not transmit an Echo Reply with IPsec ESP using already expired CHILD_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.4: Sending Multiple Transform

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	CREATE_CHILD_SA exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
   ...               ...
    |                 |                          ---
    |<----------------| IPsec {Echo Request}      |
    |                 | (Packet #3)               |
    |---------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                 | (Judgment #3)             |
    |                 |                          ---
   ...               ...
    |                 |
    |---------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                 | (Judgment #4)
    |                 |
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packets #4
Packet #3	See Common Packets #19

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired for 30 seconds.
9. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (ADVANCED)

10. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link A.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
15. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
16. Observe the messages transmitted on Link A.
17. Repeat Steps 15 and 16 until lifetime of SA is expired for 30 seconds.
18. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (ADVANCED)

19. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. TN1 responds with an IKE_SA_INIT response to the NUT.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
24. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
25. Observe the messages transmitted on Link A.
26. Repeat Steps 24 and 25 until lifetime of SA is expired for 30 seconds.
27. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Part B

Step 11: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 16: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 18: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Part C

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 25: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 27: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “No Extended Sequence Numbers” and “Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.5: Sending Multiple Proposal

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple proposals to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	CREATE_CHILD_SA exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	**ESP**	**ENCR_AES_CBC**	**AUTH_AES_XCBC_96**	**ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packets #4
Packet #3	See Common Packets #19

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired for 30 seconds.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” in SA Proposal #1 (ESP) and then “ENCR_AES_CBC”, “AUTH_AES_XCBC_96” and “Extended Sequence Numbers” in SA Proposal #2 (ESP) as accepted algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.6: Rekying Failure

Purpose:

To verify an IKEv2 device properly handles rekeying failure.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                      ---
    |<---------------| IPsec {Echo Request}   |
    |                | (Packet #3)            |
    |--------------->| IPsec {Echo Reply}     | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)          |
    |                |                      ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK(N(NO_PROPOSAL_CHOSEN))
    |                | (Packet #4)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #5)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #5)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #6)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14
Packet #5	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 reject

the NUT's proposal and responds with a CREATE_CHILD_SA response with a Notify of type NO_PROPOSAL_CHOSEN.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 13: Judgment #6
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.7: Perfect Forward Secrecy

Purpose:

To verify an IKEv2 device properly rekeys CHILD_SA when Perfect Forward Secrecy enables.

References:

- [RFC 4306] - Sections 2.12

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds. Enable PFS.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                          ---
     |<---------------| IPsec {Echo Request}      |
     |                | (Packet #3)               |
     |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)             |
     |                |                          ---
    ...              ...
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
     |                | (Judgment #4)
     |<---------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, KEr, TSi, TSr})
     |                | (Packet #4)
     |                |
     |--------------->| INFORMATIONAL request (HDR, SK {D})
     |                | (Judgment #5)
     |<---------------| INFORMATIONAL response (HDR, SK {D})
     |                | (Packet #5)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #6)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #6)
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See below
Packet #6	See Common Packet #19 This packet is cryptographically protected by the new CHILD_SA negotiated at Step 10.

Packet #4: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #14	
UDP Header	Same as the Common Packet #14	
IKEv2 Header	Same as the Common Packet #14	
E Payload	Same as the Common Packet #14	
N Payload	Same as the Common Packet #14	
N Payload	Same as the Common Packet #14	
SA Payload	Same as the Common Packet #14	
Nr Payload	Next Payload	34 (KE)
KEr Payload	Next Payload	44 (TSi)

	Critical	0
	Reserved	0
	Payload Length	136
	DH Group #	2
	Reserved	0
	Key Exchange Data	any
TSi Payload	Same as the Common Packet #14	
TSr Payload	Same as the Common Packet #14	

Packet #5: INFORMATIONAL response

IPv6 Header	Same as the Common Packet #18	
UDP Header	Same as the Common Packet #18	
IKEv2 Header	Same as the Common Packet #18	
E Payload	Other fields are same as the Common Packet #18	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response with a Delete payload to the NUT.
13. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Step 14: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.3.8: Use of the old CHILD_SA

Purpose:

To verify an IKEv2 device properly handles new CHILD_SA and old CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
   ...              ...
    |                |                            ---
    |<---------------| IPsec {Echo Request}        |
    |                | (Packet #3)                 |
    |--------------->| IPsec {Echo Reply}          | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)               |
    |                |                            ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Packet #4)
    |                |
    |<---------------| IPsec {Echo Request} (old CHILD_SA)
    |                | (Packet #5)
    |--------------->| IPsec {Echo Reply} (old CHILD_SA)
    |                | (Judgment #5)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14
Packet #5	See Common Packet #19 This packet is cryptographically protected by the new CHILD_SA negotiated at Step 5.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms again.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 12: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the first negotiated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.4. Rekeying IKE_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.EN.I.1.2.4.1: Close the replaced IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}       |
    |                | (Packet #3)                |
    |--------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)              |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Packet #4)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #5)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #5)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #6)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #6)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #12
Packet #5	See Common Packet #18
Packet #6	See Common Packet #19

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds

with a CREATE_CHILD_SA response to the NUT.

11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response to close the replaced IKE_SA.
13. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 14: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.2: Use of the new IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
 (End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                         ---
    |<---------------| IPsec {Echo Request}       |
    |                | (Packet #3)                |
    |--------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)              |
    |                |                         ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Packet #4)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #5)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #5)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {})
    |                | (Packet #6)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #6)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #12
Packet #5	See Common Packet #18
Packet #6	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds

with a CREATE_CHILD_SA response to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close the replaced IKE_SA.
13. TN1 transmits an INFORMATIONAL request with no payloads cryptographically protected by new IKE_SA.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 14: Judgment #6
The NUT resopndes with an INFORMATIONAL response with not payloads cryptographically protected by new IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.3: Lifetime of IKE_SA expires

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |<---------------| INFORMATIONAL request (HDR, SK {})
     |                | (Packet #3)
     |--------------->| INFORMATIONAL response (HDR, SK {})
     |                | (Judgment #3)
     |                |
     |                * wait for the event of a timeout of IKE_SA
     |                |
     |<---------------| INFORMATIONAL request (HDR, SK {})
     |                | (Packet #4)
     |--------X       | INFORMATIONAL response (HDR, SK {})
     |                | (Judgment #4)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #17
Packet #4	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
7. Observe the messages transmitted on Link A.
8. TN1 waits for the event of a timeout on the NUT.
9. After timeout of CHILD_SA on the NUT, TN1 transmits an INFORMATIONAL request with no payloads using already expired IKE_SA.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT responds with an INFORMATIONAL response with no payloads.

Step 10: Judgment #4
The NUT does not respond with an INFORMATIONAL response with no payloads using already expired IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.4: Sending Multiple Transform

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple transforms to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.

	CREATE_CHILD_SA exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_HMAC_SHA1** **PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 2** **Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                 | (Packet #2)
     |                 |
    ...               ...
     |                 |                        ---
     |<----------------| IPsec {Echo Request}     |
     |                 | (Packet #3)              |
     |---------------->| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
     |                 | (Judgment #3)            |
     |                 |                        ---
    ...               ...
     |                 |
     |---------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                 | (Judgment #4)
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo-Random Functions (ADVANCED)

10. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link A.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
15. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
16. Observe the messages transmitted on Link A.
17. Repeat Steps 6 and 7 until lifetime of SA is expired.
18. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (ADVANCED)

19. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. TN1 responds with an IKE_SA_INIT response to the NUT.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
24. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
25. Observe the messages transmitted on Link A.
26. Repeat Steps 6 and 7 until lifetime of SA is expired.
27. Observe the messages transmitted on Link A.

Part D: Multiple D-H Groups (ADVANCED)

28. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
29. Observe the messages transmitted on Link A.
30. TN1 responds with an IKE_SA_INIT response to the NUT.
31. Observe the messages transmitted on Link A.
32. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
33. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
34. Observe the messages transmitted on Link A.
35. Repeat Steps 6 and 7 until lifetime of SA is expired.
36. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Part B

Step 11: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 16: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 18: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "PRF_HMAC_SHA1", "PRF_AES128_CBC", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's SPI value in the SPI field.

Part C

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 25: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 27: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96", "AUTH_AES_XCBC_96" and "D-H group 2" as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's SPI value in the SPI field.

Part D

Step 29: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 31: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 34: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 36: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96", "D-H group 2" and "D-H group 14" as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's SPI value in the SPI field.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.5: Sending Multiple Proposal

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple proposal to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.

	CREATE_CHILD_SA exchanges Algorithms					
	Proposal	**Protocol ID**	**Encryption**	**PRF**	**Integrity**	**D-H Group**
Part A	Proposal #1	IKE	ENCR_ 3DES	PRF_ HMAC_SHA1	AUTH_ HMAC_SHA1_96	Group 2
	Proposal #2	**IKE**	**ENCR_ AES_CBC**	**PRF_ AES128_CBC**	**AUTH_ AES_XCBC_96**	**Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)        (End-Node)
    |                 |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
   ...               ...
    |                 |                             ---
    |<--------------| IPsec {Echo Request}         |
    |                 | (Packet #3)                  |
    |-------------->| IPsec {Echo Reply}           | repeat Echo exchange until lifetime of SA is expired
    |                 | (Judgment #3)                |
    |                 |                             ---
   ...               ...
    |                 |
    |-------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                 | (Judgment #4)
    V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” in SA Proposal #1 (ESP) and “ENCR_AES_CBC”, “PRF_AES128_CBC”, “AUTH_AES_XCBC_96” and “D-H group 14” in SA Proposal #2 (ESP) as proposed algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.6: Use of the old IKE_SA

Purpose:

To verify an IKEv2 device properly handles new CHILD_SA and old CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK SA, Ni})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Packet #4)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {}) (old IKE_SA)
    |                | (Packet #5)
    |--------------->| INFORMATIONAL response (HDR, SK {}) (old IKE_SA)
    |                | (Judgment #5)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #12
Packet #5	See Common Packet #17 (Use old IKE_SA)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is encrypted by the old IKE_SA.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #5
The NUT transmits an INFORMATIONAL response with no payload to the TN1. THe message is encrypted by the old IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.4.7: Changing PRFs when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
  Configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA Rekeying Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}       |
    |                | (Packet #3)                |
    |--------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)              |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Packet #4)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #5)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #5)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {})
    |                | (Packet #6)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #6)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See Common Packet #18
Packet #6	See Common Packet #17

Packet #4: CREATE_CHILD_SA response

Packet #4 is same as Common Packet #12 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	14 (2048 MODP Group)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close the replaced IKE_SA.
13. TN1 transmits an INFORMATIONAL request with no payloads cryptographically protected by new IKE_SA.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 14: Judgment #6
The NUT resopndes with an INFORMATIONAL response with not payloads cryptographically protected by new IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.5. Creating New CHILD_SAs with the CREATE_CHILD_SA Exchanges

Test IKEv2.EN.I.1.2.5.1: Create new CHILD_SA by sending CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to generate new CHILD_SAs.

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3
- [RFC 4718] - Sections 4.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See below
Packet #2	See Common Packet #4

Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #4	
UDP Header	Same as the Common Packet #4	
IKEv2 Header	Same as the Common Packet #4	
E Payload	Same as the Common Packet #4	
IDi Payload	Same as the Common Packet #4	
AUTH Payload	Same as the Common Packet #4	
N Payload	Same as the Common Packet #4	
SA Payload	Same as the Common Packet #4	
TSi Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. NUT starts to negotiate new CHILD_SA with TN1 by sending CREATE_CHILD_SA request.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.I.1.2.5.2: Receipt of cryptographically valid message on the new SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to generate new CHILD_SAs.

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3
- [RFC 4718] - Sections 4.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                 TN1
(End-Node)        (End-Node)
    |                   |
    |-------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                   |  (Judgment #1)
    |<--------------|  IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                   |  (Packet #1)
    |                   |
    |-------------->|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                   |  (Judgment #2)
    |<--------------|  IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   |  (Packet #2)
    |                   |
    |<--------------|  IPsec {TCP-SYN}
    |                   |  (Packet #3)
    |-------------->|  IPsec {TCP-RST}
    |                   |  (Judgment #3)
    |                   |
    |<--------------|  IPsec {Echo Request}
    |                   |  (Packet #4)
    |-------X           |  IPsec {Echo Reply}
    |                   |  (Judgment #4)
    |                   |
    |-------------->|  CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
    |                   |  (Judgment #5)
    |<--------------|  CREATE_CHILD_SA response (HDR, SK {N, SA, Nr, TSi, TSr})
    |                   |  (Packet #5)
    |                   |
    |<--------------|  IPsec {TCP-SYN}
    |                   |  (Packet #6)
    |-------------->|  IPsec {TCP-RST}
    |                   |  (Judgment #6)
    |                   |
    |<--------------|  IPsec {Echo Request}
    |                   |  (Packet #7)
    |-------------->|  IPsec {Echo Reply}
    |                   |  (Judgment #7)
    |                   |
    V                   V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packets #19
Packet #5	See below
Packet #6	See below
Packet #7	See Common Packet #19

- Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #4
UDP Header	Same as the Common Packet #4
IKEv2 Header	Same as the Common Packet #4
E Payload	Same as the Common Packet #4
IDi Payload	Same as the Common Packet #4
AUTH Payload	Same as the Common Packet #4
N Payload	Same as the Common Packet #4
SA Payload	Same as the Common Packet #4

TSi Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

- Packet #3: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

- Packet #5: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #8	
UDP Header	Same as the Common Packet #8	
IKEv2 Header	Same as the Common Packet #8	
E Payload	Same as the Common Packet #8	
IDi Payload	Same as the Common Packet #8	
AUTH Payload	Same as the Common Packet #8	
N Payload	Same as the Common Packet #8	
SA Payload	Same as the Common Packet #8	
TSi Payload	Other fields are same as the Common Packet #8	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #8	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link X
		Ending Address	NUT's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)

		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link A
		Ending Address	TN1's Global Address on Link A

- Packet #6: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port 30000 on NUT.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
9. Observe the messages transmitted on Link A.
10. NUT starts to negotiate new CHILD_SA with TN1 by sending CREATE_CHILD_SA request.
11. Observe the messages transmitted on Link A.
12. After a reception of CREATE_CHILD_SA request from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port 30000 on NUT.
14. Observe the messages transmitted on Link A.
15. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT never transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 14: Judgment #6
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 16: Judgment #7
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- If the NUT uses TCP port 30000 for other applications, the TN1 transmits TCP-SYN packets to other closed TCP port on the NUT.

Group 2.6. Exchange Collisions

Test IKEv2.EN.I.1.2.6.1: Simultaneous CHILD_SA Close

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA message to close CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
     |                |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<--------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |                * wait until expiring CHILD_SA
     |                |
     |-------------->| INFORMATIONAL request (HDR, SK {D})
     |                | (Judgment #3)
     |                |
     |<--------------| INFORMATIONAL request (HDR, SK {D})
     |                | (Packet #3)
     |                |
     |<--------------| INFORMATIONAL response (HDR, SK { })
     |                | (Packet #4)
     |                |
     |-------------->| INFORMATIONAL response (HDR, SK { })
     |                | (Judgment #4)
     |                |
     |<--------------| IPsec {Echo Request}
     |                | (Packet #5)
     |-------X        | IPsec {Echo Reply}
     |                | (Judgment #5)
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See below
Packet #4	See Common Packet #17
Packet #5	See Common Packet #19

Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0

	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request to close CHILD_SA established at Step 5.
9. TN1 responds with an INFROMATIONAL response with no payload to an INFORMATIONAL request received at Step 7.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Step 10: Judgment #4
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close CHILD_SA.

Step 12: Judgment #5

The NUT never transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.2: Simultaneous IKE_SA Close

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA message to close IKE_SA.

References:

- [RFC 4718] - Sections 5.11.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<--------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
    |                 * wait until expiring CHILD_SA
    |                 |
    |-------------->| INFORMATIONAL request (HDR, SK {D})
    |                 | (Judgment #3)
    |                 |
    |<--------------| INFORMATIONAL request (HDR, SK {D})
    |                 | (Packet #3)
    |                 |
    |<--------------| INFORMATIONAL response (HDR, SK { })
    |                 | (Packet #4)
    |                 |
    |-------------->| INFORMATIONAL response (HDR, SK { })
    |                 | (Judgment #4)
    |                 |
    |<--------------| INFORMATIONAL request (HDR, SK { })
    |                 | (Packet #5)
    |-------X         | INFORMATIONAL response (HDR, SK { })
    |                 | (Judgment #5)
    V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See below
Packet #4	See Common Packet #17
Packet #5	See Common Packet #17

Packet #3 INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request to close CHILD_SA established at Step 5.
9. TN1 responds with an INFROMATIONAL response with no payload to an

INFORMATIONAL response received at Step 7.

10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is cryptographically protected by IKE_SA to be closed.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Step 10: Judgment #4
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close CHILD_SA.

Step 12: Judgment #5
The NUT never transmits an INFORMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.3: Simultaneous CHILD_SA Rekeying

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Judgment #1)
    |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                 | (Packet #1)
    |                 |
    |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 | (Judgment #2)
    |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 | (Packet #2)
    |                 |
   ...               ...
    |                 |                          ---
    |<----------------| IPsec {Echo Request}       |
    |                 | (Packet #3)                |
    |---------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                 | (Judgment #3)              |
    |                 |                          ---
   ...               ...
    |                 |
    |---------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                 | (Judgment #4)
    |                 |
    |<----------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Nr, TSi, TSr})
    |                 | (Packet #4)
    |---------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Ni, TSi, TSr})
    |                 | (Judgment #5)
    |                 |
    |<----------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                 | (Packet #5)
    |                 |
    |---------------->| INFORMATIONAL request (HDR, SK {D})
    |                 | (Judgment #6)
    |<----------------| INFORMATIONAL response (HDR, SK {D})
    |                 | (Packet #6)
    |                 |
    |---------------->| INFORMATIONAL request (HDR, SK {D})
    |                 | (Judgment #7)
    |<----------------| INFORMATIONAL response (HDR, SK {D})
    |                 | (Packet #7)
    |                 |
    |<----------------| IPsec {Echo Request}  (new CHILD_SA)
    |                 | (Packet #8)
    |---------------->| IPsec {Echo Reply}    (new CHILD_SA)
    |                 | (Judgment #8)
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #13
Packet #5	See Common Packet #14
Packet #6	See below
Packet #7	See below
Packet #8	See Common Packet #19

Packet #6: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Packet #7: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size

	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the new CHILD_SA initiated by the NUT at Step 9

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey CHILD_SA to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with a CREATE_CHILD_SA response to the CRETE_CHILD_SA received at Step 9. The response message includes minimum Nonce Data.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response to the INFORMATIONAL request received at Step 13.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an INFORMATIONAL response to the INFORMATIONAL request received at Step 15.
17. TN1 transmits an Echo Request with IPsec ESP using the existing algorithms to the NUT.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence

Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the original CHILD_SA.

Step 15: Judgment #7
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the new CHILD_SA initiated by the NUT at Step 9.

Step 18: Judgment #8
The NUT transmits an Echo Reply with IPsec ESP using the existing CHILD_SA initiated by the TN1 at Step 10.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.4: Simultaneous CHILD_SA Rekeying with retransmission

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
(End-Node)        (End-Node)
    |                  |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                  | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                  | (Packet #1)
    |                  |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                  | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                  | (Packet #2)
    |                  |
   ...                ...
    |                  |                          ---
    |<---------------| IPsec {Echo Request}       |
    |                  | (Packet #3)                |
    |--------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                  | (Judgment #3)              |
    |                  |                          ---
   ...                ...
    |                  |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                  | (Judgment #4)
    |                  |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Nr, TSi, TSr})
    |                  | (Packet #4)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Ni, TSi, TSr})
    |                  | (Judgment #5)
    |                  |
    |<---------------| INFORMATIONAL request (HDR, SK {D})
    |                  | (Packet #5)
    |--------------->| INFORMATIONAL response (HDR, SK {D})
    |                  | (Judgment #6)
    |                  |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                  | (Judgment #7)
    |<---------------| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)}
    |                  | (Packet #6)
    |                  |
    |<---------------| IPsec {Echo Request}
    |                  | (Packet #7)
    |--------------->| IPsec {Echo Reply}
    |                  | (Judgment #8)
    V                  V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #13
Packet #5	See below
Packet #6	See below
Packet #7	See Common Packet #19

Packet #5: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500

	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Packet #6: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #14
UDP Header		Same as Common Packet #14
IKEv2 Header		Same as Common Packet #14
E Payload		Same as Common Packet #14
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	NO_PROPOSAL_CHOSEN (14)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey CHILD_SA to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an INFORMAITONAL request with a Delete Payload to close the replaced

CHILD_SA.

13. Observe the messages transmitted on Link A.
14. Observe the messages transmitted on Link A.
15. TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type NO_PROPOSAL_CHOSEN to the retransmitted CREATE_CHILD_SA request.
16. TN1 transmits an Echo Request with IPsec ESP using the existing algorithms to the NUT.
17. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL response with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the original CHILD_SA.

Step 14: Judgment #7
The NUT retransmits the same CREATE_CHILD_SA request as the message at Step 11. The message includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 17: Judgment #8
The NUT transmits an Echo Reply with IPsec ESP using the existing CHILD_SA initiated by the TN1 at Step 10.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.5: Simultaneous IKE_SA Rekeying

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4718] - Sections 5.11.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                         ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                         ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Packet #4)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Judgment #5)
    |                |
    |<---------------| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                | (Packet #5)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #6)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #6)
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #7)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #7)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {})
    |                | (Packet #8)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #8)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #11
Packet #5	See Common Packet #12
Packet #6	See Common Packet #18
Packet #7	See Common Packet #18
Packet #8	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with a CREATE_CHILD_SA response to the CREATE_CHILD_SA request received at Step 9. The response message includes minimum Nonce Data to make the NUT send a message to close duplicated IKE_SA.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response with no payload.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an INFORMATIONAL response with no payload.
17. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is cryptographically protected by the new IKE_SA initiated by TN1 at Step 10.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT responds a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the proposal in the SA payload Response has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s responder’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request . The message's IKE_SA Initiator's SPI value is the IKE_SA Initiator's SPI value of the original IKE_SA, and the message's IKE_SA Responder's SPI value is the IKE_SA Responder's SPI value of the original IKE_SA. The message also has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Step 15: Judgment #7
The NUT transmits an INFORMATIONAL request . The message's IKE_SA Initiator's SPI value is the IKE_SA Initiator's SPI value of the new IKE_SA initiated by the NUT at Step 9, and the message's IKE_SA Responder's SPI value is the IKE_SA Responder's SPI value of the new IKE_SA initiated by the NUT at Step 9. The message also has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Step 18: Judgment #8
The NUT transmits an INFOMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.6: Simultaneous IKE_SA Rekeying with retransmission

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4718] - Sections 5.11.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
    ...              ...
     |                |                        ---
     |<---------------| IPsec {Echo Request}    |
     |                | (Packet #3)             |
     |--------------->| IPsec {Echo Reply}      | repeat Echo exchange until lifetime of SA is expired
     |                | (Judgment #3)           |
     |                |                        ---
    ...              ...
     |                |
     |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                | (Judgment #4)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                | (Packet #4)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
     |                | (Judgment #4)
     |                |
     |<---------------| INFORMATIONAL request (HDR, SK {D})
     |                | (Packet #5)
     |--------------->| INFORMATIONAL response (HDR, SK {})
     |                | (Judgment #5)
     |                |
     |-------X        | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                | (Judgment #6)
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #11
Packet #5	See below

Packet #5: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0

	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an INFORMATONAL request to close the original IKE_SA. The message has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.
13. Observe the messages transmitted on Link A.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT responds a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the proposal in the SA payload Response has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA's responder's SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds with an INFOMATIONAL response to the INFORMATIONAL request to close the original IKE_SA.

Step 14: Judgment #7
The NUT never retransmits a CREATE_CHILD_SA request transmitted at Step 9.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.7: Rekeying a CHILD_SA while Closing a CHILD_SA

Purpose:

To verify an IKEv2 device properly handles simultaneous closing and rekeying a CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)      (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |                * wait until CHILD_SA expires
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #3)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |                |
    |<---------------| INFORMATIONAL response (HDR, SK {D})
    |                | (Packet #4)
    |                |
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
    |                | (Judgment #4)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Packet #3	See Common Packet #13
Packet #4	See below

Packet #4: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey a CHILD_SA.
8. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close a CHILD_SA.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATONAL request to close a CHILD_SA.

Step 9: Judgment #4
The NUT responds with a CREATE_CHILD_SA response to a CREATE_CHILD_SA reqest to rekey a CHILD_SA. The CREATE_CHILD_SA response includes a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.8: Closing a New CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a request to close nonexistent CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.6

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                        ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                        ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |        X-------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Packet #4)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {D})
    |                | (Packet #5)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #5)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14
Packet #5	See below

Packet #5: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any

E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA response to rekey a CHILD_SA to the NUT. But the response does not reach the NUT.
11. TN1 transmits an INFORMATIONAL request to close a CHILD_SA which were supposed to be created by rekey.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 12: Judgment #5
The NUT responds with an INFORMATIONAL response with no payload to the TN1.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.9: Rekeying a New CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a request to rekey nonexistent CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                           ---
    |<---------------| IPsec {Echo Request}       |
    |                | (Packet #3)                |
    |--------------->| IPsec {Echo Reply}         | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)              |
    |                |                           ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |        X-------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Packet #4)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #5)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
    |                | (Judgment #5)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14
Packet #5	See Common Packet #14 The SPI value in the Delete payload is the same value as the SPI value in Packet #4 SA payload.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA response to rekey a CHILD_SA to the NUT. But the

response does not reach the NUT.

11. TN1 transmits a CREATE_CHILD_SA request to rekey the CHILD_SA which were supposed to be created by rekey.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 12: Judgment #5
The NUT responds with a CREATE_CHILD_SA response with a Notify of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.10: Rekeying an IKE_SA with half-open CHILD_SAs

Purpose:

To verify an IKEv2 device properly handles a request to rekey an IKE_SA which has CHILD_SAs in half-open state.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                          ---
    |<---------------| IPsec {Echo Request}      |
    |                | (Packet #3)               |
    |--------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)             |
    |                |                          ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Judgment #4)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Packet #4)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
    |                | (Judgment #5)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA to the NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 11: Judgment #5
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.11: Rekeying a CHILD_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to rekey a CHILD_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                        ---
    |<---------------| IPsec {Echo Request}    |
    |                | (Packet #3)             |
    |--------------->| IPsec {Echo Reply}      | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)           |
    |                |                        ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #4)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_ADDITIONAL_SAS)})
    |                | (Judgment #5)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #13

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits a CREATE_CHILD_SA request to rekey a CHILD_SA to the NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_ADDTIONAL_SAS to a CREATE_CHILD_SA request to rekey a CHILD_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.12: Rekeying an IKE_SA with half-closed CHILD_SAs

Purpose:

To verify an IKEv2 device properly handles a request to rekey an IKE_SA which has CHILD_SAs in half-closed state.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
 (End-Node)        (End-Node)
     |                 |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                 | (Packet #2)
     |                 |
     |                 * wait until CHILD_SA expires
     |                 |
     |-------------->| INFORMATIONAL request (HDR, SK {D})
     |                 | (Judgment #3)
     |                 |
     |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                 | (Packet #3)
     |-------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
     |                 | (Judgment #4)
     |                 |
     V                 V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA to the NUT.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATIONAL request to close a CHILD_SA to the TN1.

Step 8: Judgment #4
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.13: Closing a CHILD_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to close a CHILD_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
   ...              ...
    |                |                        ---
    |<---------------| IPsec {Echo Request}    |
    |                | (Packet #3)             |
    |--------------->| IPsec {Echo Reply}      | repeat Echo exchange until lifetime of SA is expired
    |                | (Judgment #3)           |
    |                |                        ---
   ...              ...
    |                |
    |--------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Judgment #4)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {D})
    |                | (Packet #4)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below

Packet #4: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits an INFORMATIONAL request to close a CHILD_SA to the NUT.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT responds with an INFORMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.14: Closing an IKE_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to close an IKE_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
     NUT                 TN1
  (End-Node)          (End-Node)
       |                   |
       |------------------>| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
       |                   | (Judgment #1)
       |<------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
       |                   | (Packet #1)
       |                   |
       |------------------>| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
       |                   | (Judgment #2)
       |<------------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
       |                   | (Packet #2)
       |                   |
      ...                 ...
       |                   |                         ---
       |<------------------| IPsec {Echo Request}     |
       |                   | (Packet #3)              |
       |------------------>| IPsec {Echo Reply}       | repeat Echo exchange until lifetime of SA is expired
       |                   | (Judgment #3)            |
       |                   |                         ---
      ...                 ...
       |                   |
       |------------------>| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
       |                   | (Judgment #4)
       |                   |
       |<------------------| INFORMATIONAL request (HDR, SK {D})
       |                   | (Packet #4)
       |                   |
       |<------------------| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
       |                   | (Packet #5)
       |                   |
       |------------------>| INFORMATIONAL response (HDR, SK {})
       |                   | (Judgment #5)
       |                   |
       V                   V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See below

Packet #4: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0

	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Packet #5: CREATE_CHILD_SA response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	14 (NO_PROPOSAL_CHOSEN)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. TN1 transmits an INFORMATIONAL request to close an IKE_SA to the NUT.
11. TN1 responds with a CREATE_CHILD_SA response which has a Notify payload of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 12: Judgment #5
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.I.1.2.6.15: Rekeying an IKE _SA while Closing an IKE_SA

Purpose:

To verify an IKEv2 device properly handles simultaneous closing and rekeying an IKE_SA.

References:

- [RFC 4718] - Sections 5.11.10

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Packet #2)
    |                |
    |                * wait until CHILD_SA expires
    |                |
    |-------------->| INFORMATIONAL request (HDR, SK {D})
    |                | (Judgment #3)
    |                |
    |<--------------| CREATE_CHILD_SA request (HDR, SK { SA, Ni })
    |                | (Packet #3)
    |                |
    |-------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
    |                | (Judgment #4)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATONAL request to close an IKE_SA.

Step 8: Judgment #4
The NUT responds with a CREATE_CHILD_SA response to a CREATE_CHILD_SA reqest to rekey an IKE_SA. The CREATE_CHILD_SA response includes a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.7. Non zero RESERVED fields

Test IKEv2.EN.I.1.2.7.1: Non zero RESERVED fields in CREATE_CHILD_SA response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
 (End-Node)        (End-Node)
     |                 |
     |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Judgment #1)
     |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 | (Packet #1)
     |                 |
     |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                 | (Judgment #2)
     |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                 | (Packet #2)
     |                 |
    ...               ...
     |                 |                        ---
     |<----------------| IPsec {Echo Request}      |
     |                 | (Packet #3)               |
     |---------------->| IPsec {Echo Reply}        | repeat Echo exchange until lifetime of SA is expired
     |                 | (Judgment #3)             |
     |                 |                        ---
    ...               ...
     |                 |
     |---------------->| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                 | (Judgment #4)
     |<----------------| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
     |                 | (Packet #4)
     |                 |
     |---------------->| INFORMATIONAL request (HDR, SK {D})
     |                 | (Judgment #5)
     V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19
Packet #4	See Common Packet #14 All RESERVED fields are set to one.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link A.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT. All RESERVED fields in the message are set to one.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 3. The INFORMATIONAL Exchange

Group 3.1. Header and Payload Formats

Test IKEv2.EN.I.1.3.1.1: Sending INFORMATIONAL Exchange

Purpose:

To verify an IKEv2 device checks whether the other endpoint is alive.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |               * wait until receiving a liveness check
    |               |
    |-------------->| INFORMATIONAL request (HDR, SK {})
    |               | (Judgment #3)
    |               |
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
9. Observe the messages transmitted on Link A.
10. TN1 responds with an IKE_SA_INIT response to the NUT.
11. Observe the messages transmitted on Link A.
12. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
13. TN1 waits for receiving an INFORMATIONAL request with no payloads.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 35 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.

- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to INFORMATIONAL (37).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to the value incremented the previous IKE message's Message ID by one.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 9: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 1: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 14: Judgment #3
The NUT transmits an INFORMATIONAL request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 36 Encrypted payload

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.

- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Possible Problems:

- None

Group 3.2. Use of Retransmission Timers

Test IKEv2.EN.I.1.3.2.1: Retransmission of INFORMATIONAL request

Purpose:

To verify an IKEv2 device properly retransmits INFORMATIONAL request

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |                * wait until receiving liveness check
     |                |
     |--------------->| INFORMATIONAL request (HDR, SK {})
     |                | (Judgment #3)
     |                |
     |                * wait for the event of a timeout
     |                |
     |--------------->| INFORMATIONAL request (HDR, SK {})
     |                | (Judgment #4)
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link B.
5. TN1 waits for reception of IKE_AUTH response from the NUT.
6. TN1 waits for reception of INFROMATIONAL request for liveness check from the NUT.
7. Observe the messages transmitted on Link B.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it. And the request has the same Message ID value as the Message ID value received at Step 7.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.EN.I.1.3.2.2: Stop of retransmission of INFORMATIONAL request

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |                * wait until receiving liveness check
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {})
    |                | (Judgment #3)
    |                |
    |                * wait for the event of a timeout
    |                |
    |--------------->| INFORMATIONAL request (HDR, SK {})
    |                | (Judgment #4)
    |<---------------| INFORMATIONAL response (HDR, SK {})
    |                | (Packet #3)
    |                |
    |                * wait for the event of a timeout
    |                |
    |-------X        | never send INFORMATIONAL request (HDR, SK {})
    |                | (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Packet #2	See Common Packet #4
Packet #3	See Common Packet #18

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link B.
5. TN1 waits for reception of IKE_AUTH response from the NUT.
6. TN1 transmits an Echo Request with invalid SPI.
7. Observe the messages transmitted on Link B.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link B.
10. After reception of an INFORMATIONAL request from the NUT, TN1 responds with an INFORMATIONAL response to the NUT.
11. TN1 waits for the event of a timeout on NUT.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it. And the request has the same Message ID value as the request received at Step 7.

Step 12: Judgment #5
The NUT never retransmits an INFORMATIONAL request which has the same Message ID value as the received Step 9.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Group 3.3. Non zero RESERVED fields

Test IKEv2.EN.I.1.3.3.1: Non zero RESERVED fields in INFORMATIONAL response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime and CHILD_SA Lifetime to more than twice as INFORMATIONAL message retransmission timer as.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |               * wait until receiving a liveness check
    |               |
    |-------------->| INFORMATIONAL request (HDR, SK {})
    |               | (Judgment #3)
    |<--------------| INFORMATIONAL response (HDR, SK {})
    |               | (Packet #3)
    |               |
    |-------X       | INFORMATIONAL request (HDR, SK {})
    |               | (Judgment #3)
    |               |
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2

Packet #2	See Common Packet #4
Packet #3	See Common Packet #18 All RESERVED fields are set to one.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link A.
8. TN1 responds with an INFORMATIONAL response with no payload to the NUT. All RESERVED fields in the message are set to one.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT never retransmit an INFORMATIONAL request.

Possible Problems:

- None

Group 3.4. Error Handling

Test IKEv2.EN.I.1.3.4.1: INVALID_SPI

Purpose:

To verify an IKEv2 device properly handles ESP packet with invalid SPI.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #3)
     |                |
     |--------------->| INFORMATIONAL request (HDR, SK {N(INVALID_SPI)})
     |                | (Judgment #3)
     |                |
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #19 This packet has an invalid SPI value (the properly netotiated value plus 1).

Part A (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms. The message’s SPI is set to the value of the SPI negotiated in the initial exchange plus 1.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Notify payload of type INVALID_SPI. The Notify payload includes the SPI value which is transmitted at Step 6.

Possible Problems:

- None.

Section 1.1.2. Endpoint to Security Gateway Tunnel

Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.EN.I.2.1.1.1: Sending IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)         (SGW)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                | (Packet #1)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V
```

Packet #1	See Common Packet #2

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response to the NUT.

8. Observe the messages transmitted on Link A.

Part C: IDi Payload Format (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.

Part D: AUTH Payload Format (BASIC)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link A.

Part E: SA Payload Format (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.

Part F: TSi Payload Format (BASIC)

21. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link A.
23. TN1 responds with an IKE_SA_INIT response to the NUT.
24. Observe the messages transmitted on Link A.

Part G: TSr Payload Format (BASIC)

25. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. TN1 responds with an IKE_SA_INIT response to the NUT.
28. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 37 Header format

- An IKE_SA Initiator's SPI field is set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field is set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to IKE_AUTH (35).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to 1.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 38 Encrypted payload

- A Next Payload field is set to IDi Payload (35).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 39 ID Payload format

- A Next Payload field is set to AUTH Payload (39).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field is set to ID_IPV6_ADDR (5).
- A RESERVED field is set to zero.
- An Identification Data field is set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 40 AUTH Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field is set to Shared Key Message Integrity Code (2).
- A RESERVED field is set to zero.
- An Authentication Data field is set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2

```
                              1                   2                   3
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
         ! Next    44   !0!     0       ! Length          40           !              |
         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---           |
         !      0       !       0       ! Length          36           !  |           |
         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
         ! Number     1 ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3 !  |           |
         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
         ! SPI value                                                   !  |           |
     --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
        | !      3       !       0       ! Length           8           !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |SA Payload
        | ! Type  1  (EN) !       0       ! Transform ID     3     (3DES) !  | Proposal  |
     --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
        | !      3       !       0       ! Length           8           !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
        | ! Type  3  (IN) !       0       ! Transform ID     2     (SHA1) !  |           |
     --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
        | !      0       !       0       ! Length           8           !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
        | ! Type  5  (ESN)!       0       ! Transform ID     0       (No)  !  |           |
     --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 41 SA Payload contents

The NUT transmits an IKE_AUTH request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 42 SA Payload format

- A Next Payload field is set to TSi Payload (44).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 43 Proposal sub-structure format

Transform field is set to following (There are 3 Transform Structures).

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field is set to 1 if this structure is the first proposal, otherwise set to 1 greater thatn the previous proposal.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to 4.
- A # of Transforms field is set to 3.
- A SPI field is set to the sending entity's SPI (4 octets value)

Transform field is set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 44 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field is set to zero if this structure is the last proposal, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field is set to ESN (5).
- A RESERVED field is set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 24: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 45 TSi Payload format

- A Next Payload field is set to TSr Payload (45).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to the number of actual traffic selectors.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 46 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to NUT address.
- A Ending Address field is set to greater thatn or equal to NUT address.

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 47 TSr Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to the number of actual traffic selectors.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                      1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !   TS Type     !IP Protocol ID*|       Selector Length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Start Port*         |           End Port*           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !                                                               !
 ~                         Starting Address*                     ~
 !                                                               !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !                                                               !
 ~                         Ending Address*                       ~
 !                                                               !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 48 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to less than or equal to Prefix Y.
- An Ending Address field is set to less than or equal to Prefix Y.

Possible Problems:

- IKE_AUTH request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDi,
[CERT+],
[N(INITIAL_CONTACT)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
[IDr],
AUTH,
[CP(CFG_REQUEST)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA,
TSi,
TSr,
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

- Each of transforms can be located in the any order.

Test IKEv2.EN.I.2.1.1.2: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1              TH1
(End-Node)          (SGW)            (Host)
     |                |                 |
     |-------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                |                 | (Judgment #1)
     |<--------------|                 | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                |                 | (Packet #1)
     |                |                 |
     |-------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
     |                |                 | (Judgment #2)
     |<--------------|                 | IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
     |                |                 | (Packet #2)
     |                |                 |
     |<==============+-----------------| IPsec {Echo Request}
     |                |                 | (Packet #3)
     |===============+---------------->| IPsec {Echo Reply}
     |                |                 | (Judgment #3)
     |                |                 |
     V                V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #20

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH1 transmits an Echo Request and TN1 forwards an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Group 1.2. Requesting an Internal Address on a Remote Network

Test IKEv2.EN.I.2.1.2.1: Sending CFG_REQUEST

Purpose:

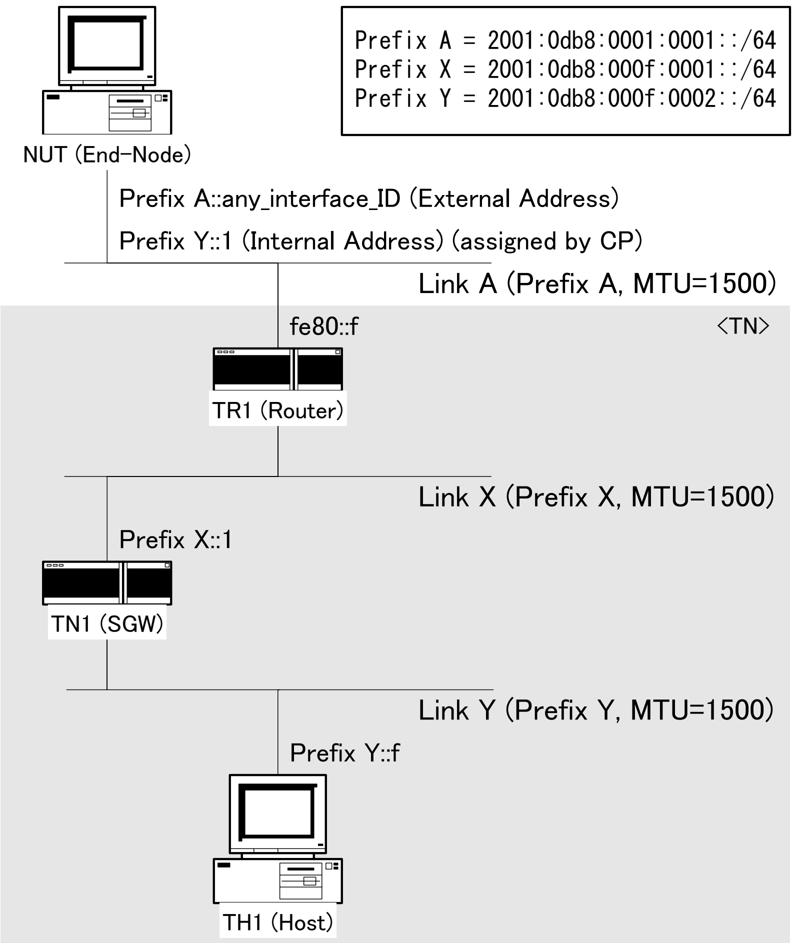
To verify an IKEv2 device transmits IKE_AUTH request using properly Configuration Payload format

References:

- [RFC 4306] - Sections 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REQUEST for INTERNAL_IP6_ADDRESS. The traffic selector must be configured by the following

table.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT (internal address)	ANY	ANY
Outbound	NUT (internal address)	ANY	ANY	Link Y	ANY	ANY

* NUT must propose Traffic Selector covering above address range.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)          (SGW)
     |                  |
     |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Judgment #1)
     |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                  | (Packet #1)
     |                  |
     |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, CP(CFG_REQUEST), SAi2, TSi, TSr})
     |                  | (Judgment #2)
     |                  |
     V                  V
```

Packet #1	See Common Packet #2

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response to the NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted Configuration Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C! RESERVED    !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   CFG Type    !                    RESERVED                   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Configuration Attributes                    ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 49 Configuration Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A CFG Type field is set to CFG_REQUEST (1).
- A RESERVED field is set to zero.

The following configuration attribute must be included in Configuration Attributes field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!R|         Attribute Type      !            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                             Value                             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 50 Configuration Attributes format

Configuration Attribute #1

- Reserved field is set to zero.
- Attribute Type field is set to INTERNAL_IP6_ADDRESS (8).
- Length field is set to zero.
- Value field is empty.

Possible Problems:

- The implementation may not set single configuration attribute by the implementation policy. In this case, Configuration Payload contains multiple configuration attributes.

Test IKEv2.EN.I.2.1.2.2: Receipt of CFG_REPLY

Purpose:

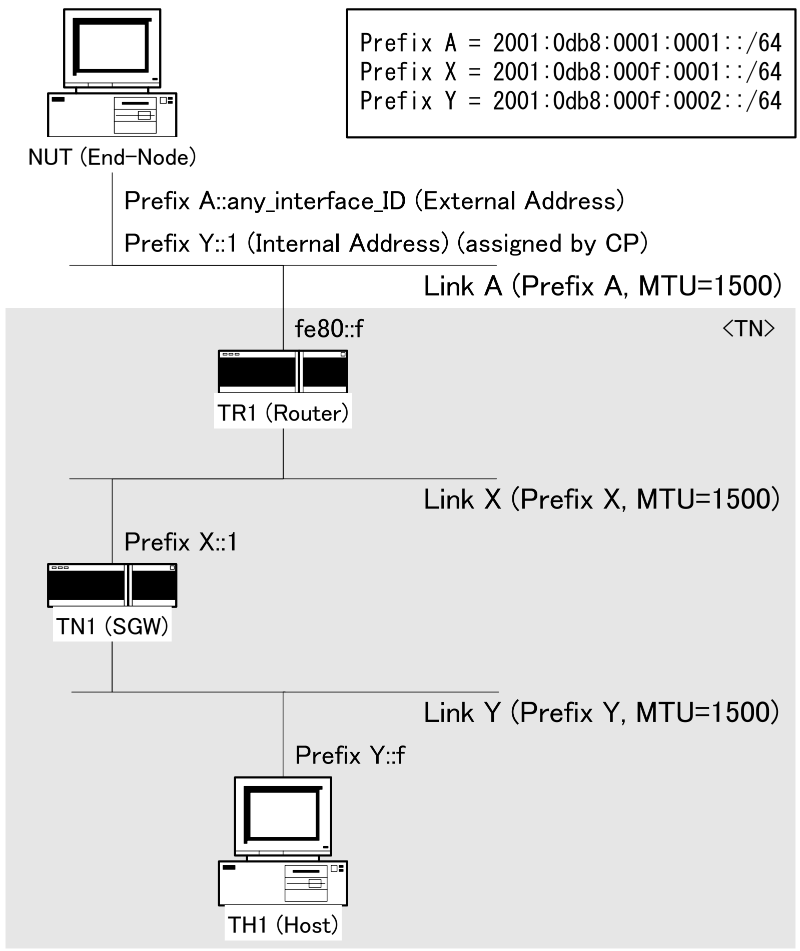
To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 2.19 and 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REQUEST for INTERNAL_IP6_ADDRESS. The traffic selector must be configured by the following table.

	Traffic Selector	
	Source	**Destination**

	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT (internal address)	ANY	ANY
Outbound	NUT (internal address)	ANY	ANY	Link Y	ANY	ANY

* NUT must propose Traffic Selector covering above address range.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1               TH1
 (End-Node)          (SGW)            (Host)
     |                 |                 |
     |-------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 |                 | (Judgment #1)
     |<--------------|                 | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                 |                 | (Packet #1)
     |                 |                 |
     |-------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH,
     |                 |                 |                CP(CFG_REQUEST), SAi2, TSi, TSr})
     |                 |                 | (Judgment #2)
     |<--------------|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
     |                 |                 | (Packet #2)
     |                 |                 |
     |<==============+---------------| IPsec {Echo Request (sent to NUT internal address)}
     |                 |                 | (Packet #3)
     |==============+--------------->| IPsec {Echo Reply (sent from NUT internal address)}
     |                 |                 | (Judgment #3)
     |                 |                 |
     V                 V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Below
Packet #3	See Below

- Packet #2: IKE_AUTH response packet

IPv6 Header	Same as Common Packet #6	
UDP Header	Same as Common Packet #6	
IKEv2 Header	Same as Common Packet #6	
E Payload	Same as Common Packet #6	
IDr Payload	Same as Common Packet #6	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #6	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	29
	CFG Type	2 (CFG_REPLY)
	RESERVED	0
	Configuration Attributes	See below
SA Payload	Same as Common Packet #6	
TSi Payload	Other fields are same as Common Packet #6	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #6	

Configuration Attributes	Reserved	0

	Attribute Type	INTERNAL_IP6_ADDRESS	
	Length		17
	Value	IPv6 address	Prefix Y::1
		Prefix-length	128

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	Prefix Y::1
	Ending Address	Prefix Y::1

- Packet #3: Echo Request packet

IPv6 Header	Same as Common Packet #20	
ESP	Same as Common Packet #20	
IPv6 Header	Source Address	Prefyx Y::f
	Destination Address	Prefix Y::1
ICMPv6 Header	Same as Common Packet #20	

Part A (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH1 transmits an Echo Request to NUT internal address and TN1 forwards an Echo Request with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96. The inner packet is sent from NUT internal address.

Possible Problems:

- None.

Test IKEv2.EN.I.2.1.2.3: Non zero RESERVED fields in Configuration Payload

Purpose:

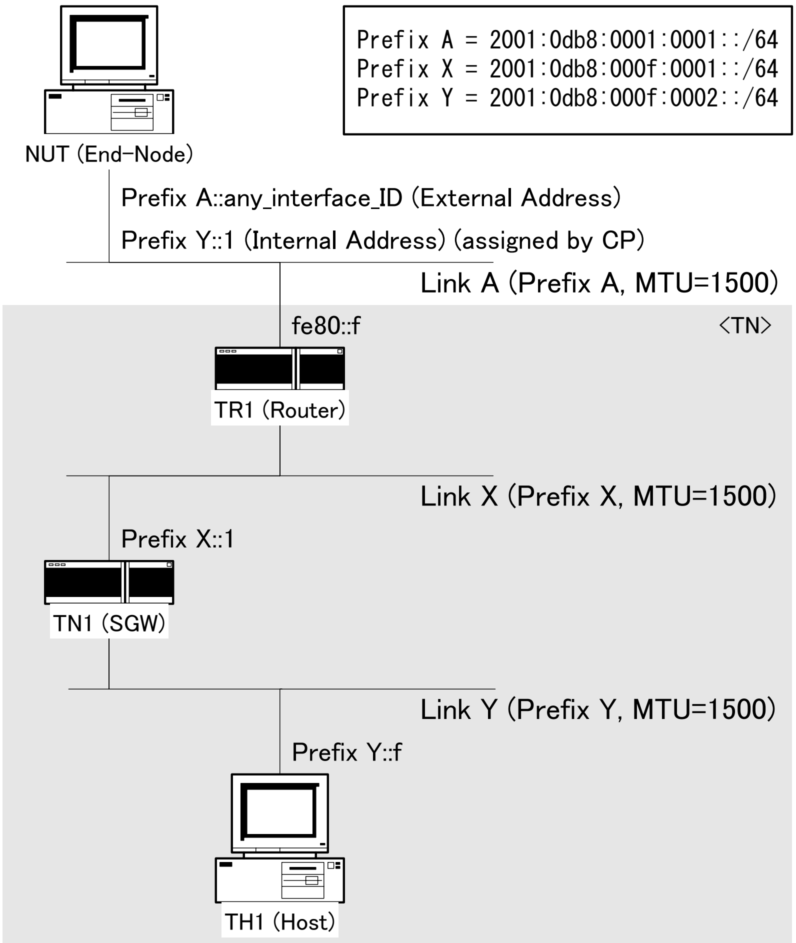
To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REQUEST for INTERNAL_IP6_ADDRESS. The traffic selector must be configured by the following table.

	Traffic Selector	
	Source	**Destination**

	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT (internal address)	ANY	ANY
Outbound	NUT (internal address)	ANY	ANY	Link Y	ANY	ANY

* NUT must propose Traffic Selector covering above address range.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                TN1               TH1
(End-Node)          (SGW)             (Host)
   |                  |                 |
   |-------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                  |                 | (Judgment #1)
   |<--------------|                 | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                  |                 | (Packet #1)
   |                  |                 |
   |-------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH,
   |                  |                 |                  CP(CFG_REQUEST), SAi2, TSi, TSr})
   |                  |                 | (Judgment #2)
   |<--------------|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
   |                  |                 | (Packet #2)
   |                  |                 |
   |<==============+--------------| IPsec {Echo Request (sent to NUT internal address)}
   |                  |                 | (Packet #3)
   |==============+-------------->| IPsec {Echo Reply (sent from NUT internal address)}
   |                  |                 | (Judgment #3)
   |                  |                 |
   V                  V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Below
Packet #3	See Below

- Packet #2: IKE_AUTH response packet

IPv6 Header	Same as Common Packet #6	
UDP Header	Same as Common Packet #6	
IKEv2 Header	Same as Common Packet #6	
E Payload	Same as Common Packet #6	
IDr Payload	Same as Common Packet #6	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #6	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	1
	Payload Length	29
	CFG Type	2 (CFG_REPLY)
	RESERVED	1
	Configuration Attributes	See below
SA Payload	Same as Common Packet #6	
TSi Payload	Other fields are same as Common Packet #6	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #6	

Configuration Attributes	Reserved	1

	Attribute Type	INTERNAL_IP6_ADDRESS	
	Length	17	
	Value	IPv6 address	Prefix Y::1
		Prefix-length	128

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	Prefix Y::1
	Ending Address	Prefix Y::1

- Packet #3: Echo Request packet

IPv6 Header	Same as Common Packet #20	
ESP	Same as Common Packet #20	
IPv6 Header	Source Address	Prefyx Y::f
	Destination Address	Prefix Y::1
ICMPv6 Header	Same as Common Packet #20	

Part A (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH1 transmits an Echo Request to NUT internal address and TN1 forwards an Echo Request with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96. The inner packet is sent from NUT internal address.

Possible Problems:

- None.

Test IKEv2.EN.I.2.1.2.4: Receipt of IKE_AUTH response without CFG_REPLY

Purpose:

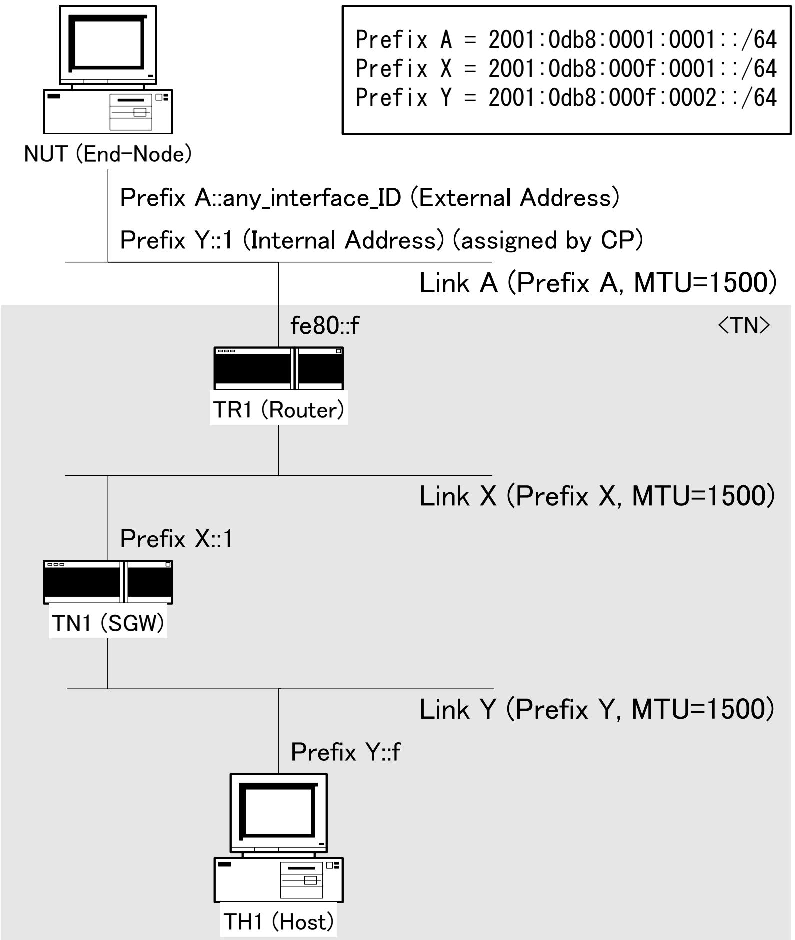
To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4718] - Sections 6.8

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REQUEST for INTERNAL_IP6_ADDRESS. The traffic selector must be configured by the following table.

	Traffic Selector	
	Source	**Destination**

	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT (internal address)	ANY	ANY
Outbound	NUT (internal address)	ANY	ANY	Link Y	ANY	ANY

* NUT must propose Traffic Selector covering above address range.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
(End-Node)            (SGW)
     |                  |
     |----------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Judgment #1)
     |<-----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                  | (Packet #1)
     |                  |
     |----------------->| IKE_AUTH request (HDR, SK {IDi, AUTH,
     |                  |                 CP(CFG_REQUEST), SAi2, TSi, TSr})
     |                  | (Judgment #2)
     |<-----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
     |                  | (Packet #2)
     |                  |
     |<-----------------| INFORMATIONAL request (HDR, SK {})
     |                  | (Packet #3)
     |----------------->| INFORMATIONAL response (HDR, SK {})
     |                  | (Judgment #3)
     V                  V
```

Packet #1	See Common Packet #2
Packet #2	See Below
Packet #3	See Common Packet #17

- Packet #2: IKE_AUTH response packet

IPv6 Header	Same as Common Packet #6	
UDP Header	Same as Common Packet #6	
IKEv2 Header	Same as Common Packet #6	
E Payload	Same as Common Packet #6	
IDr Payload	Same as Common Packet #6	
AUTH Payload	Next Payload	33 (SA)
	Other fields are same as Common Packet #6	
SA Payload	Same as Common Packet #6	
TSi Payload	Other fields are same as Common Packet #6	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #6	

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	Prefix Y::1
	Ending Address	Prefix Y::1

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT. The message does not include any Configuration payloads.
6. TH1 transmits an INFORMATIONAL request with no payload to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with no payload to the TN1.

Possible Problems:

- None.

Test IKEv2.EN.I.2.1.2.5: Receipt of unrecognized Configuration Attributes

Purpose:

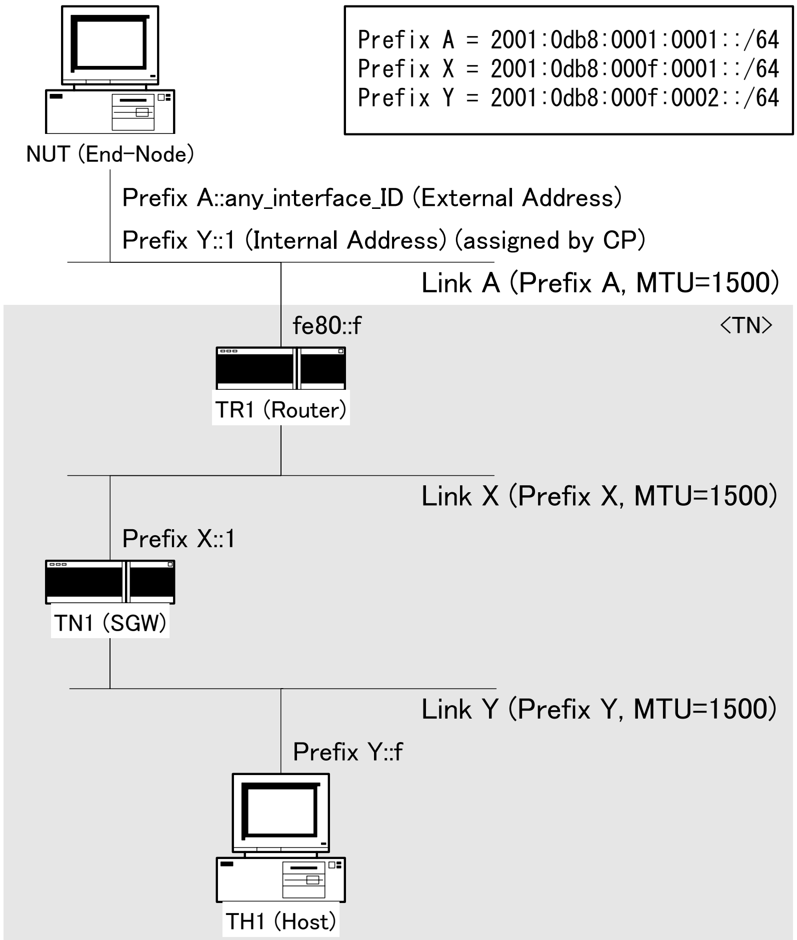
To verify an IKEv2 device properly handles unrecognized Configuration Attributes.

References:

- [RFC 4306] - Sections 2.19 and 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REQUEST for INTERNAL_IP6_ADDRESS. The traffic selector must be configured by the following table.

	Traffic Selector	
	Source	Destination

	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	Link Y	ANY	ANY	NUT (internal address)	ANY	ANY
Outbound	NUT (internal address)	ANY	ANY	Link Y	ANY	ANY

* NUT must propose Traffic Selector covering above address range.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(End-Node)           (SGW)
  |                     |
  |-------------->|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                     |  (Judgment #1)
  |<--------------|  IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                     |  (Packet #1)
  |                     |
  |-------------->|  IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                     |                          CP(CFG_REQUEST), SAi2, TSi, TSr})
  |                     |  (Judgment #2)
  |<--------------|  IKE_AUTH Response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
  |                     |  (Packet #2)
  |                     |
  |<--------------|  INFORMATIONAL request (HDR, SK {})
  |                     |  (Packet #3)
  |-------------->|  INFORMATIONAL response (HDR, SK {})
  |                     |  (Judgment #3)
  V                     V
```

Packet #1	See Common Packet #2
Packet #2	See Below
Packet #3	See Common Packet #17

- Packet #2: IKE_AUTH response packet

IPv6 Header	Same as Common Packet #6	
UDP Header	Same as Common Packet #6	
IKEv2 Header	Same as Common Packet #6	
E Payload	Same as Common Packet #6	
IDr Payload	Same as Common Packet #6	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #6	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	29
	CFG Type	2 (CFG_REPLY)
	RESERVED	0
	Configuration Attributes	See below
SA Payload	Same as Common Packet #6	
TSi Payload	Other fields are same as Common Packet #6	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #6	

Configuration Attributes	Reserved	0
	Attribute Type	32767

	Length	17	
	Value	IPv6 address	Prefix Y::1
		Prefix-length	128

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	Prefix Y::1
	Ending Address	Prefix Y::1

Part A (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT. The message includes a Configuration Attribute of unrecognized Attribute Type.
6. TH1 transmits an INFORMATIONAL request with no payload to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with no payload to the TN1.

Possible Problems:

- None.

Section 1.2. Responder

Section 1.2.1. Endpoint-to-Endpoint Transport

Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.EN.R.1.1.1.1: Sending IKE_SA_INIT response

Purpose:

To verify an IKEv2 device transmits an IKE_SA_INIT response using properly Header and Payloads format

References:

- [RFC4306] - Section 1.2, 2.10, 3.1, 3.2, 3.3, 3.4 and 3.9
- [RFC 4718] - Sections 7.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT            TN1
(End-Node)     (End-Node)
    |              |
    |<-------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |              | (Packet #1)
    |------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |              | (Judgment #1)
    V              V
```

Packet #1	See Common Packet #1

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Part B: SA Payload Format (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
4. Observe the messages transmitted on Link A.

Part C: KE Payload Format (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.

Part D: Nonce Payload Format (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 51 Header format

- An IKE_SA Initiator’s SPI field is set to IKE_SA Initiator’s SPI field value supplied in the first IKE_SA_INIT request message.
- An IKE_SA Responder’s SPI field is set to a 64-bits value chosen by the NUT. It MUST not be zero.
- A Next Payload field is set to SA Payload (33).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to IKE_SA_INIT (34).
- A Flags field is set to (00000100)2 = (4)10.
- A Message ID field is set to zero.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 4: Judgment #1

```
                         1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
    ! Next     34   !0!      0      ! Length         44             !             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---         |
    !       0       !        0      ! Length         40             !  |          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
    ! Number     1  ! Prot ID    1  ! SPI Size  0   ! Trans Cnt  4  !  |          |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | !       3       !        0      ! Length          8             !  |          |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | ! Type  1  (EN) !        0      ! Transform ID     3     (3DES) !  |          |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | !       3       !        0      ! Length          8             !  |          |SA Payload
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |Proposal  |
  | ! Type  2  (PR) !        0      ! Transform ID     2     (SHA1) !  |          |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | !       3       !        0      ! Length          8             !  |          |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | ! Type  3  (IN) !        0      ! Transform ID     2     (SHA1) !  |          |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | !       0       !        0      ! Length          8             !  |          |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
  | ! Type  4  (DH) !        0      ! Transform ID     2     (1024) !  |          |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 52 SA Payload contents

The NUT transmits an IKE_SA_INIT response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 53 SA Payload format

- A Next Payload field is set to KE Payload (34).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 54 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 40 bytes for this proposal according to Common Configuration.
- A Proposal # field is set to 1.
- A Protocol ID field is set to IKE (1).
- A SPI Size field is set to zero.
- A # of Transforms field is set to 4.

A Transform field is set to following (There are 4 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 55 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.

- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for PRF_HMAC_SHA1.
- A Transform Type field is set to PRF (2).
- A RESERVED field is set to zero.
- A Transform ID set to PRF_HMAC_SHA1 (2).

Transform #3
- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #4
- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for 1024 MODP Group.
- A Transform Type field is set to D-H (4).
- A RESERVED field is set to zero.
- A Transform ID set to Group2 (2).

Part C

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including properly formatted KE Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          DH Group #           !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Key Exchange Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 56 KE Payload format

- A Next Payload field is set to Nonce Payload (40).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 136 bytes for Group 2.
- A DH Group field is set to Group2 (2).
- A RESERVED field is set to zero.
- A Key Exchange Data field is set to Diffie-Hellman public value. The length of

the Key Exchange Data field must be equal to 1024bit.

- The length of the Key Exchange Data field must be equal to 1024bit.

Part D

Step 8: Judgment #4

The NUT transmits an IKE_SA_INIT response including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 57 Nonce Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Nonce Data field is set to random data generated by the transmitting entity.
- The size of the Nonce must between 16 and 256 octets.

Possible Problems:

- IKE_SA_INIT response has following packet format. It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
SA, KE, Nr,
[N(NAT_DETECTION_SOURCE_IP),
 N(NAT_DETECTION_DESTINATION_IP)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)],
  CERTREQ+],
[V+]
```

- Each of transforms can be located in the any order.

Test IKEv2.EN.R.1.1.1.2: Sending IKE_AUTH response

Purpose:

To verify an IKEv2 device transmits an IKE_AUHT response using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
8. Observe the messages transmitted on Link A.

Part C: IDr Payload Format (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
12. Observe the messages transmitted on Link A.

Part D: AUTH Payload Format (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.

Part E: Notify Payload Format (BASIC)

17. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
20. Observe the messages transmitted on Link A.

Part F: SA Payload Format (BASIC)

21. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
24. Observe the messages transmitted on Link A.

Part G: TSi Payload Format (BASIC)

25. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
28. Observe the messages transmitted on Link A.

Part H: TSr Payload Format (BASIC)

29. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
30. Observe the messages transmitted on Link A.
31. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
32. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 58 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.
- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to IKE_AUTH (35).
- A Flags field is set to $(00000100)_2 = (4)_{10}$.
- A Message ID field is set to 1.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 59 Encrypted payload

- A Next Payload field is set to IDr Payload (36).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 60 ID Payload format

- A Next Payload field is set to AUTH Payload (39).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field is set to ID_IPV6_ADDR (5).
- A RESERVED field is set to zero.
- An Identification Data field is set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 61 AUTH Payload format

- A Next Payload field is set to Notify Payload (41).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1
- An Auth Method field is set to Shared Key Message Integrity Code (2).
- A RESERVED field is set to zero.
- An Authentication Data field is set to correct authentication value.

Part E

Step 18: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 20: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 62 Notify Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 8 bytes for USE_TRANSPORT.
- A Protocol ID field is set to IKE_SA (1).
- A SPI Size field is set to zero.
- A Notify Message Type field is set to USE_TRANSPORT_MODE (16391)

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 24: Judgment #2

```
                            1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
       ! Next     44    !0!      0      ! Length          40            !            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
       !       0       !        0       ! Length          36            !  |         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
       ! Number     1   ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3  !  |         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
       ! SPI value                                                      !  |         |
   --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
     | !       3       !        0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |SA Payload
     | ! Type  1  (EN) !        0       ! Transform ID     3     (3DES) !  | Proposal |
   --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
     | !       3       !        0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
     | ! Type  3  (IN) !        0       ! Transform ID     2     (SHA1) !  |         |
   --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
     | !       0       !        0       ! Length           8            !  |         |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |         |
     | ! Type  5  (ESN)!        0       ! Transform ID     0       (No) !  |         |
   --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 63 SA Payload contents

The NUT transmits an IKE_AUTH response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 64 SA Payload format

- A Next Payload field is set to TSi Payload (44).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 65 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field is set to 1.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to 4.
- A # of Transforms field is set to 3.
- A SPI field is set to the sending entity's SPI (4 octets value)

Transform field is set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 66 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.

* A RESERVED field is set to zero.
* A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
* A Transform Type field is set to INTEG (3).
* A RESERVED field is set to zero.
* A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3
* A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
* A RESERVED field is set to zero.
* A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
* A Transform Type field is set to ESN (5).
* A RESERVED field is set to zero.
* A Transform ID set to No Extended Sequence Numbers (0).

Part G

Step 26: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 28: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 67 TSi Payload format

* A Next Payload field is set to TSr Payload (45).
* A Critical field is set to zero.
* A RESERVED field is set to zero.
* A Payload Length field is set to length of the current payload.
* A Number of TSs field is set to 1.
* A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 68 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to TN1 address.
- An Ending Address field is set to TN1 address.

Part H

Step 30: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 32: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 69 TSr Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to 1.
- A RESERVED field is set to zero.

Traffic Selectors field is set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 70 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to NUT address.
- An Ending Address field is set to NUT address.

Possible Problems:

- IKE_AUTH response has following packet format. It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDr,
[CERT+],
AUTH,
[CP(CFG_REPLY)],
[N(IPCOMP_SUPPORTED)],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA,
TSi,
TSr,
[N(ADDITIONAL_TS_POSSIBLE)],
[V+]
```

- Each of transforms can be located in the any order.

Test IKEv2.EN.R.1.1.1.3: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles CHILD_SA negotiated by the Initial Exchanges using Pre-shared key.

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
(End-Node)         (End-Node)
    |                   |
    |<------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                   | (Packet #1)
    |------------------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                   | (Judgment #1)
    |                   |
    |<------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                   | (Packet #2)
    |------------------>| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   | (Judgment #2)
    |                   |
    |<------------------| IPsec {Echo Request}
    |                   | (Packet #3)
    |------------------>| IPsec {Echo Reply}
    |                   | (Judgment #3)
    V                   V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.

6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.2. Use of Retransmission Timers

Test IKEv2.EN.R.1.1.2.1: Receipt of retransmitted IKE_SA_INIT request

Purpose:

To verify an IKEv2 device transmits an IKE_SA_INIT response when the device received a retransmitted IKE_SA_INIT request.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |                * wait until retrans timer expires
    |-------X        | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #2)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #2)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #3)
    |                |
    V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1 (The Message ID is the same as Packet #1)

Part A: (BASIC)

1. TN starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. Observe the messages transmitted on Link A.
4. TN1 retransmits same IKE_SA_INIT request as the message transmitted in Step 1 to the

NUT.

5. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 3: Judgment #2
The NUT never retransmits the same IKE_SA_INIT response as the response transmitted at Step 2.

Step 5: Judgment #3
The NUT transmits the same IKE_SA_INIT response as the response transmitted at Step 2.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.2.2: Receipt of retransmitted IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits an IKE_AUTH response when the device received a retransmitted IKE_AUTH request.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
(End-Node)         (End-Node)
    |                   |
    |<------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                   | (Packet #1)
    |------------------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                   | (Judgment #1)
    |                   |
    |<------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                   | (Packet #2)
    |------------------>| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   | (Judgment #2)
    |                   |
    |                   * wait until retrans timer expires
    |--------X          | IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   | (Judgment #3)
    |                   |
    |<------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                   | (Packet #3)
    |------------------>| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                   | (Judgment #4)
    |                   |
    V                   V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #3 (The Message ID is the same as Packet #1)

Part A: (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. After reception of an IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. Observe the messages transmitted on Link A.
6. TN1 retransmits the same IKE_AUTH request as the request transmitted in Step 3 to the NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 5: Judgment #3
The NUT never retransmits the same IKE_AUTH response as the response transmitted at Step 4.

Step 7: Judgment #4
The NUT transmits the same IKE_AUTH response as the response transmitted at Step 4.

Possible Problems:

- None.

Group 1.3. State Synchronization and Connection Timeouts

Test IKEv2.EN.R.1.1.3.1: State Synchronization with ICMP messages

Purpose:

To verify that an IKEv2 device doesn't conclude that the other endpoint has faild by receiving ICMP Error messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT               TR1               TN1
(End-Node)        (Router)          (End-Node)
   |                 |                 |
   |<----------------+-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 |                 | (Packet #1)
   |-----------------+---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                 |                 | (Judgment #1)
   |                 |                 |
   |<----------------+-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
   |                 |                 | (Packet #2)
   |-----------------+---------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
   |                 |                 | (Judgment #2)
   |                 |                 |
   |<----------------+-----------------| IPsec {Echo Request}
   |                 |                 | (Packet #3)
   |-----------------+---------------->| IPsec {Echo Reply}
   |                 |                 | (Judgment #3)
   |                 |                 |
   |<----------------|                 | Destination Unreachable (No route to destination)
   |                 |                 | (Packet #4)
   |                 |                 |
   |<----------------+-----------------| IPsec {Echo Request}
   |                 |                 | (Packet #5)
   |-----------------+---------------->| IPsec {Echo Reply}
   |                 |                 | (Judgment #4)
   |                 |                 |
   V                 V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1

Packet #2	See Common Packet #3
Packet #3	See Common Packet #19
Packet #4	See below
Packet #5	See Common Packet #19

- Packet #4: ICMPv6 Destination Unreachable

IPv6 Header	Source Address	TR1's Global Address on Link A
	Destination Address	NUT's Global Address on Link A
ICMPv6 Header	Type	1
	Code	0

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. After reception of an Echo Reply from NUT, TR1 transmits ICMP Destination Unreachable Message to the NUT.
8. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.3.2: State Synchronization with IKE messages

Purpose:

To verify that an IKEv2 device doesn't conclude that the other endpoint has faild by receiving cryptographicaly unprotected IKE message.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #3)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #3)
    |                |
    |<---------------| cryptographically unprotected IKE message
    |                | (Packet #4)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #5)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #4)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19
Packet #4	See below

Packet #5	See Common Packet #19

- Packet #4: cryptographicaly unprotected INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link A
	Destination Address	NUT's Global Address on Link X
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. After reception of an Echo Reply from NUT, TN1 transmits a cryptographically unprotected INFORMATIONAL request with Notify payload of type INVALID_ SPI to the NUT.
8. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None

Test IKEv2.EN.R.1.1.3.3: Close connections when receiving INITIAL_CONTACT

Purpose:

To verify an IKEv2 device closes connections when receiving INITIAL_CONTACT.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 7.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(End-Node)         (End-Node)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #1)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #1)
     |                  |
     |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                  | (Packet #2)
     |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                  | (Judgment #2)
     |                  |
     |<-----------------| IPsec {Echo Request}
     |                  | (Packet #3)
     |----------------->| IPsec {Echo Reply}
     |                  | (Judgment #3)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #4)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #4)
     |                  |
     |<-----------------| IKE_AUTH request (HDR, SK {IDi, N(INITIAL_CONTACT), AUTH, N, SAi2, TSi, TSr})
     |                  | (Packet #5)
     |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                  | (Judgment #5)
     |                  |
     |<-----------------| IPsec {Echo Request} (old CHILD_SA)
     |                  | (Packet #6)
     |-------X          | IPsec {Echo Reply}
     |                  | (Judgment #6)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni) (new CHILD_SA)
     |                  | (Packet #7)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #7)
     |                  |
     V                  V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19
Packet #4	See Common Packet #1
Packet #5	See Common Packet #3
Packet #6	See Common Packet #19 This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 4.
Packet #7	See Common Packet #19 This packet is cryptographically protected by the CHILD_SA negotiated at Step 7 to Step 10.

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. After reception of an Echo Reply from NUT, TN1 transmits IKE_SA_INIT request to the NUT.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH response with a Notify payload of type INITIAL_CONTACT to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithm.
12. Observe the messages transmitted on Link A.
13. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithm.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #5
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #6
The NUT never transmits an Echo Reply using the first negotiated algorithms or the second negotiated algorithms.

Step 14: Judgment #7
The NUT transmits an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.3.4: Receiving Liveness check

Purpose:

To verify that an IKEv2 device can respond to INFORMATIONAL request for liveness check.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT             TN1
(End-Node)      (End-Node)
     |               |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |               | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |               | (Judgment #1)
     |               |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |               | (Packet #2)
     |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |               | (Judgment #2)
     |               |
     |<--------------| INFORMATIONAL request (HDR, SK {})
     |               | (Packet #3)
     |-------------->| INFORMATIONAL response (HDR, SK {})
     |               | (Judgment #3)
     |               |
     V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #17

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an

INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Test IKEv2.EN.R.1.1.3.5: Receiving Delete Payload for IKE_SA

Purpose:

To verify an IKEv2 device can respond to INFORMATIONAL request with a Delete Payload, when IKE_SA is deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
(End-Node)         (End-Node)
     |                 |
     |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Packet #1)
     |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                 | (Judgment #1)
     |                 |
     |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Packet #2)
     |---------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                 | (Judgment #2)
     |                 |
     |<----------------| INFORMATIONAL request (HDR, SK {D})
     |                 | (Packet #3)
     |---------------->| INFORMATIONAL response (HDR, SK {})
     |                 | (Judgment #3)
     |                 |
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any

	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	2
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with a Delete payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with no payloads.

Possible Problems:

- None

Test IKEv2.EN.R.1.1.3.6: Receiving Delete Payload for CHILD_SA

Purpose:

To verify an IKEv2 device can respond to INFORMATIONAL request with a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
(End-Node)         (End-Node)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #1)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #1)
     |                  |
     |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                  | (Packet #2)
     |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                  | (Judgment #2)
     |                  |
     |<-----------------| INFORMATIONAL request (HDR, SK {D})
     |                  | (Packet #3)
     |----------------->| INFORMATIONAL response (HDR, SK {D})
     |                  | (Judgment #3)
     |                  |
     V                  V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any

	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	2
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with a Delete payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the TN1's inbound SPI value to be deleted as SPI value.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with a Delete payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the NUT's inbound SPI value to be deleted as SPI value.

Possible Problems:

- None

Group 1.4. Version Numbers and Forward Compatibility

Test IKEv2.EN.R.1.1.4.1: Receipt of a higher minor version number

Purpose:

To verify an IKEv2 device aceepts a request with a higher minor version number and respond to the request.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |  (Packet #1)
    |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |  (Judgment #1)
    V                V
```

Packet #1	See below

- Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Other fields are same as the Common Packet #1	
	Major Version	2
	Minor Version	1
SA Payload	Same as the Common Packet #1	
KE Payload	Same as the Common Packet #1	
Ni, Nr Payload	Same as the Common Packet #1	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request with a higher minor version number.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.4.2: Receipt of a higher major version number

Purpose:

To verify an IKEv2 device drops a request with a higher major version number and send a notification message.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
 (End-Node)        (End-Node)
      |                |
      |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
      |                |  (Packet #1)
      |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
      |                |  (Judgment #1)
      |                |
      |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
      |                |  (Packet #2)
      |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
      |                |  (Judgment #2)
      |                |
      |<---------------|  INFORMATIONAL request (HDR, SK{})
      |                |  (Packet #3)
      |-------X        |  No Response
      |                |    or
      |--------------->|  INFORMATIONAL response (HDR, SK{N(INVALID_MAJOR_VERSION)}
      |                |  (Judgment #3)
      V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: INFORMATIONAL response packet

IPv6 Header	Same as the common packet #17	
UDP Header	Same as the common packet #17	
IKEv2 Header	Other fields are same as the common packet #17	
	Major Version	3
	Minor Version	0

E Payload	Same as the common packet #17

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with a higher major version number to the NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type INVALID_MAJOR_VERSION containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 71 Notify Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A SPI Size field is set to zero.
- A Notify Message Type field is set to INVALID_MAJOR_VERSION (5).
- A Notification Data field is set to the highest version number it supports (2).

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.4.3: Unrecognized payload types and critical bit is not set

Purpose:

To verify an IKEv2 device ignores invalid payload types when the invalid type payload's critical bit is not set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {P, N, N+, SA, Ni, TSi, TSr})
     |                | (Packet #3)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
     |                | (Judgment #3)
     |                |
     V                V

P: Payload with an invalid payload type
N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	All fields are same as Common Packet #13 Payload	
UDP Header	All fields are same as Common Packet #13 Payload	
IKEv2 Header	All fields are same as Common Packet #13 Payload	
E Payload	Next Payload	*Invalid payload type value*

	Other fields are same as Common Packet #13	
Invalid Payload	Next Payoad	41 (N)
	Critical	0
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #13 Payload	
N Payload	All fields are same as Common Packet #13 Payload	
SA Payload	All fields are same as Common Packet #13 Payload	
Ni, Nr Paylaod	All fields are same as Common Packet #13 Payload	
TSi Paylaod	All fields are same as Common Packet #13 Payload	
TSr Payload	All fields are same as Common Packet #13 Payload	

Part A: Invalid payload type 1 (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 1 and the invalid payload's critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 32 and the invalid payload's critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 49 and the invalid payload's critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload

type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.4.4: Unrecognized payload types and critical bit is set

Purpose:

To verify an IKEv2 device drops invalid payload types when the invalid type payload's critical bit is set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |  (Packet #1)
    |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |  (Judgment #1)
    |                |
    |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                |  (Packet #2)
    |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                |  (Judgment #2)
    |                |
    |<---------------|  CREATE_CHILD_SA request (HDR, SK {P, N, N+, SA, Ni, TSi, TSr})
    |                |  (Packet #3)
    |-------X        |  No Response
    |                |    or
    |--------------->|  CREATE_CHILD_SA response (HDR, SK {N(UNSUPPORTED_CRITICAL_PAYLOAD)})
    |                |  (Judgment #3)
    |                |
    V                V

P: Payload with an invalid payload type
N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	All fields are same as Common Packet #13 Payload
UDP Header	All fields are same as Common Packet #13 Payload

IKEv2 Header	All fields are same as Common Packet #13 Payload	
E Payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #13	
Invalid Payload	Next Payoad	41 (N)
	Critical	1
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #13 Payload	
N Payload	All fields are same as Common Packet #13 Payload	
SA Payload	All fields are same as Common Packet #13 Payload	
Ni, Nr Paylaod	All fields are same as Common Packet #13 Payload	
TSi Paylaod	All fields are same as Common Packet #13 Payload	
TSr Payload	All fields are same as Common Packet #13 Payload	

Part A: Invalid payload type 1 and Critical bit is set (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 1 and the pointed pyaload's Critical bit is set.
6. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 and Critical bit is set (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 32 and the pointed pyaload's Critical bit is set.
12. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 and Critical bit is set (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 49 and the pointed pyaload's Critical bit is set.
18. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 and Critical bit is set (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.

22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 255 and the pointed pyaload's Critical bit is set.
24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an CREATE_CHILD_SA response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (1).

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT doest not transmit any packets or transmits an CREATE_CHILD_SA response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (32).

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 18: Judgment #3
The NUT doest not transmit any packets or transmits an CREATE_CHILD_SA response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (49).

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3

The NUT doest not transmit any packets or transmits an CREATE_CHILD_SA response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (255).

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.4.5: Invalid Order Payloads

Purpose:

To verify an IKEv2 device properly handles IKE message with invalid order payloads.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)        (End-Node)
     |                 |
     |<----------------| IKE_SA_INIT request (HDR, SAi1, Ni, KEi)
     |                 | (Packet #1)
     |-------X         | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                 | (Judgment #1)
     |                 |
     V                 V
```

Packet #1	See Common Packet #1 KEi payload and Ni payload replace each other.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT never transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Group 1.5. Cookies

Test IKEv2.EN.R.1.1.5.1: Cookies

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT response with a Notify payload of type COOKIE.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
 (End-Node)        (End-Node)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #1)
     |       ...        | ...
     |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #2)
     |                  |
     |----------------->| IKE_SA_INIT response (HDR, N(COOKIE))
     |                  | (Judgment #1)
     |                  |
     |<-----------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
     |                  | (Packet #3)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #2)
     V                  V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1
Packet #3	See below

- Packet #3: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)

N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: Notify payload of type Cookie Format (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE with the cookie data supplied by NUT
5. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1

The NUT transmits an IKE_SA_INIT response including a IKE Header which contains zero as IKE_SA Responder's SPI field and a Notify payload of type COOKIE containing following values:.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 72 Notify Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A SPI Size field is set to zero.
- A Notify Message Type field is set to COOKIE (16390).
- A Notification Data field is set to the cookie data.

Step 5: Judgment #2

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.5.2: Invalid Cookies

Purpose:

To verify an IKEv2 device handles IKE_SA_INIT request with an invalid cookie data.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |       ...      | ...
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #2)
    |                |
    |-------------->| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Judgment #1)
    |                |
    |<--------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi, Ni)
    |                | (Packet #3)
    |-------------->| IKE_SA_INIT response (HDR, N(COOKIE"))
    |                | (Judgment #2)
    V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1
Packet #3	See below

- Packet #3: IKE_SA_INIT request packet

IPv6 Header		Same as the common packet #1
UDP Header		Same as the common packet #1
IKEv2 Header		Other fields are same as the common packet #1
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0

	Notify Message Type	COOKIE (16390)
	Notification Data	The difference value than COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE with a cookie data unexpected by NUT.
5. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1
The NUT transmits an IKE_SA_INIT response including an IKE Header which contains zero as IKE_SA Responder’s SPI field and a Notify payload of type COOKIE.

Step 5: Judgment #2
The NUT transmits an IKE_SA_INIT response including an IKE Header which contains zero as IKE_SA Responder’s SPI field and a Notify payload of type COOKIE with a new cookie data.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.5.3: Interaction of COOKIE and INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device handles interaction of COOKIE and INVALID_KE_PAYLOAD.

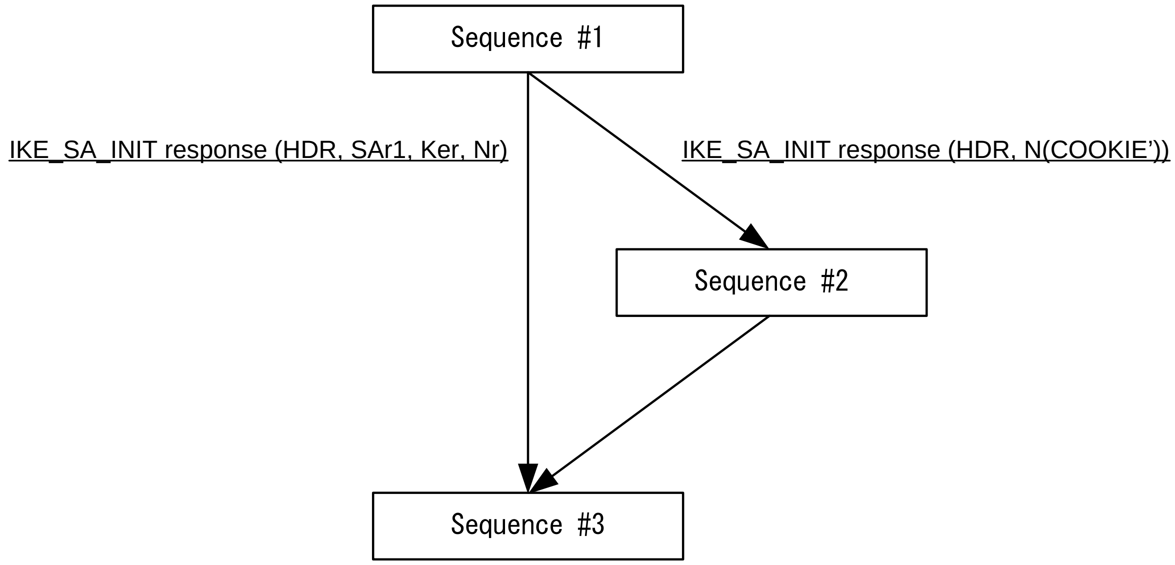
References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:



```
Sequence #1:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |       ...      | ...
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #2)
    |--------------->| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Packet #3)
    |--------------->| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Judgment #2)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni)
    |                | (Packet #4)
    |------*1------->| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                |   or
    |------*2------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #3)
    |                |
    *1) If the NUT send IKE_SA_INIT response (HDR, N(COOKIE')), go to Sequence #2.
    *2) If the NUT send IKE_SA_INIT response (HDR, SAr1, KEr, Nr), go to Sequence #3.
    Otherwise, this test is failed.

Sequence #2:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi ' , Ni)
    |                | (Packet #5)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #4)
    |                |
    Go to Sequence #3.

Sequence #3:
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #6)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See below
Packet #5	See below
Packet #6	See Common Packet #3

- Packet #1: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #2: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #3: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #4: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

- Packet #5: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1
UDP Header	Same as the common packet #1

IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT. The IKE_SA_INIT requests include an invalid KE payload which has different .DH Group # from proposing DH Group #.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has an invalid KE payload.
5. Observe the messages transmitted on Link A.
6. After reception of IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD, TN1 transmits an IKE_SA_INIT request with a valid KE payload.
7. Observe the messages transmitted on Link A.
8. If the IKE_SA_INIT response includes a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT request has a valid KE payload.
   A) Observe the messages transmitted on Link A
9. TN1 transmits an IKE_AUTH request.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 5: Judgment #2
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type INVALID_KE_PAYLOAD.

Step 7: Judgment #3
The NUT transmits an IKE_SA_INIT response. The message can contain zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE. The message can containding "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 8A: Judgment #4

The message can containding “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #5
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.5.4: Interaction of COOKIE and INVALID_KE_PAYLOAD with unoptimized Initiator

Purpose:

To verify an IKEv2 device handles interaction of COOKIE and INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |       ...      | ...
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #2)
    |                |
    |--------------->| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Packet #3)
    |--------------->| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Judgment #2)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi', Ni)
    |                | (Packet #4)
    |--------------->| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                | (Judgment #3)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KE'i, Ni)
    |                | (Packet #5)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #4)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #6)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #5)
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #1
Packet #5	See below
Packet #6	See Common Packet #3

- Packet #1: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #2: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #3: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #4: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0

	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

- Packet #5: IKE_SA_INIT request packet

IPv6 Header		Same as the common packet #1
UDP Header		Same as the common packet #1
IKEv2 Header		Other fields are same as the common packet #1
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT. The IKE_SA_INIT requests include an invalid KE payload which has different .DH Group # from proposing DH Group #.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has an invalid KE payload.
5. Observe the messages transmitted on Link A.
6. After reception of IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD, TN1 transmits an IKE_SA_INIT request with a valid KE payload.
7. Observe the messages transmitted on Link A.
8. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has a valid KE payload.
9. Observe the messages transmitted on Link A.
10. TN1 transtmits an IKE_AUTH request.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1

The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 5: Judgment #2

The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type INVALID_KE_PAYLOAD.

Step 7: Judgment #3
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 9: Judgment #4
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 11: Judgment #5
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Possible Problems:

- None.

Group 1.6. Cryptographic Algorithm Negotiation

Test IKEv2.EN.R.1.1.6.1: Cryptographic Algorithm Negotiation for IKE_SA

Purpose:

To verify an IKEv2 device properly handles various algorithms for IKE_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part E, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	***ENCR_AES_CBC***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	***ENCR_AES_CTR***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	***PRF_AES128_CBC***	AUTH_HMAC_SHA1_96	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	***AUTH_AES_XCBC_96***	Group 2
Part E	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3

Packet #1: IKE_SA_INIT requet

Packet #1 is same as Common Packet #1 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part B:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		13 (AES_CTR)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part C:

SA Transform of Tranform Type PRF is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0
	Transform Length	8
	Transform Type	2 (PRF)
	Reserved	0
	Transform ID	4 (AES128_XCBC)

Part D:

SA Transform of Tranform Type INTEG is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0
	Transform Length	8
	Transform Type	3 (INTEG)
	Reserved	0
	Transform ID	5 (AES_XCBC_96)

Part E:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	14 (2048 MODP Group)

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A.

3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
8. Observe the messages transmitted on Link A.

Part C: PRF PRF_AES128_CBC (ADVANCED)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
12. Observe the messages transmitted on Link A.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
16. Observe the messages transmitted on Link A.

Part E: D-H Group Group 14 (ADVANCED)

17. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
18. Observe the messages transmitted on Link A.
19. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
20. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_AES_CTR”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_AES128_CBC”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part D

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_AES_XCBC_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part G

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as accepted algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.2: Cryptographic Algorithm Negotiation for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles various algorithms for CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

From part A to part F, TN1 transmits an IKE_AUTH request including a SA payload which contains the transforms as follows:

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**Extended Sequence Numbers**
Part A	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part B	**ENCR_AES_CTR**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part C	**ENCR_NULL**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part D	ENCR_3DES	**AUTH_AES_XCBC_96**	No Extended Sequence Numbers
Part E	ENCR_3DES	**NONE**	No Extended Sequence Numbers
Part F	ENCR_3DES	AUTH_HMAC_SHA1_96	**Extended Sequence Numbers**

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #3)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #3)
     |                |
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1

Packet #2	See below
Packet #3	See Common Packet #19

Packet #3: IKE_SA_INIT requet

Packet #3 is same as Common Packet #3 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part B:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		13 (AES_CTR)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part C:

SA Transform of Tranform Type INTEG is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0
	Transform Length	8
	Transform Type	1 (ENCR)
	Reserved	0
	Transform ID	11 (ENCR_NULL)

Part D:

SA Transform of Tranform Type INTEG is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0
	Transform Length	8
	Transform Type	3 (INTEG)
	Reserved	0
	Transform ID	5 (AES_XCBC_96)

Part E:

SA Transform of Tranform Type INTEG is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0
	Transform Length	8
	Transform Type	3 (INTEG)
	Reserved	0
	Transform ID	0 (NONE)

Part F:

SA Transform of Tranform Type ESN is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	5 (ESN)
	Reserved	0
	Transform ID	1 (Extended Sequence Numbers)

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Encryption Algorithm ENCR_NULL (ADVANCED)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
16. Observe the messages transmitted on Link A.
17. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
24. Observe the messages transmitted on Link A.

Part E: Integrity Algorithm NONE (ADVANCED)

25. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.

28. Observe the messages transmitted on Link A.
29. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
30. Observe the messages transmitted on Link A.

Part F: Extended Sequence Numbers (ADVANCED)

31. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
32. Observe the messages transmitted on Link A.
33. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
34. Observe the messages transmitted on Link A.
35. After reception of IKE_AUTH response from the NUT, TN transmits an Echo Request with IPsec ESP with the accepted cryptographic suite to the NUT.
36. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_AES_CTR”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_NULL”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part D

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_AES_XCBC_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 24: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part E

Step 26: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 28: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "NONE" and "No Extended Sequence Numbers" as accepted algorithms.

Step 30: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Part F

Step 32: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 36: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "Extended Sequence Numbers" as accepted algorithms.

Step 38: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.3: Receiving Multiple Transforms for IKE_SA

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with an multiple transforms.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                  |
    |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                  | (Packet #1)
    |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                  | (Judgment #1)
    V                  V
```

Packet #1	See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the transforms as follows:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_AES_CBC** **ENCR_3DES**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_AES128_CBC** **PRF_HMAC_SHA1**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96** **AUTH_HMAC_SHA1_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14** **Group 2**

- Packet #1 IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Same as the Common Packet #1	
SA Payload	Other fields are same as the common packet #1	
	SA Proposals	See SA Table below

KE Payload	Same as the Common Packet #1
Ni, Nr Payload	Same as the Common Packet #1

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo-Random Functions (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A.

Part D: Multiple D-H Groups (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.

8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.4: Receiving Multiple Proposals for IKE_SA

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with multiple proposals.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     V                V
```

Packet #1	See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the proposals as follows:

	IKE_SA_INIT exchanges Algorithms					
	Proposals	Protocol ID	Encryption	PRF	Integrity	D-H Group
Part A	Proposal #1	IKE	**ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	Proposal #1	IKE	ENCR_3DES	**PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96**	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part D	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14**
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

- Packet #1 IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1
UDP Header	Same as the Common Packet #1
IKEv2 Header	Same as the Common Packet #1
SA Payload	Other fields are same as the common packet #1

	SA Proposals	See SA Table below
KE Payload	Same as the Common Packet #1	
Ni, Nr Payload	Same as the Common Packet #1	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		2
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0

			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo-Random Functions (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A.

Part D: Multiple D-H Groups (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.5: Receiving Multiple Transforms for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles an IKE_AUTH request with multiple transforms.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below

From part A to part C, TN1 transmits an IKE_AUTH request including a SA payload which contains the transforms as follows:

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Other fields are same as the Common Packet #3	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
8. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.6: Receiving Multiple Proposals for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles an IKE_AUTH request with multiple proposals.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)       (End-Node)
     |                |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below

TN1 transmits an IKE_AUTH request including a SA payload which contains the two proposals as follows:

	IKE_AUTH exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part B	Proposal #1	ESP	ENCR_3DES	**AUTH_AES_XCBC_96**	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part C	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	**ESN**
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Other fields are same as the Common Packet #3	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		2
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0

			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
8. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.7: Sending INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles an invalid KE payload which has different D-H Group # from proposed D-H Group #.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. Enable PFS.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
    |                | (Packet #3)
    |-------X        | CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, KEr, TSi, TSr})
    |                |   or
    |--------------->| CREATE_CHILD_SA response (HDR, SK { N(INVALID_KE_PAYLOAD)})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request for rekeying CHILD_SA

IPv6 Header	Same as the Common Packet #13

UDP Header	Same as the Common Packet #13	
IKEv2 Header	Same as the Common Packet #13	
E Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
SA Payload	Same as the Common Packet #13	
Ni, Nr Payload	Other fields are same as the Common Packet #13	
	Next Payload	34 (KE)
KEi Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	264
	DH Group #	14
	Reserved	0
	Key Exchange Data	any
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs. The CREATE_CHILD_SA contains a D-H Group transform to use D-H Group 2 and a Key Exchange payload which contains 14 (D-H Group 14) as DH Group # field and the Key Exchage Data.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits any packets or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_KE_PAYLOAD which contains 2 (D-H Group 2) as Notification Data.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.8: Sending INVALID_KE_PAYLOAD in Initial Exchange

Purpose:

To verify an IKEv2 device properly handles an invalid KE payload which has different D-H Group # from proposed D-H Group #.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT                TN1
(End-Node)         (End-Node)
     |                  |
     |<-----------------| IKE_SA_INIT Request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #1)
     |----------------->| IKE_SA_INIT Response (HDR, N(INVALID_KE_PAYLOAD))
     |                  | (Judgment #1)
     |                  |
     |<-----------------| IKE_SA_INIT Request (HDR, SAi1, KEi, Ni)
     |                  | (Packet #2)
     |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                  | (Judgment #2)
     |                  |
     V                  V
```

Packet #1	See below
Packet #2	See Common packet #1

- Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Same as the Common Packet #1	
SA Payload	Same as the Common Packet #1	
KEi Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	264
	DH Group #	14
	Reserved	0
	Key Exchange Data	any
Ni, Nr Payload	Same as the Common Packet #1	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload which contains a D-H Group transform proposes using D-H Group 2 and a Key Exchange payload which contains 14 (D-H Group 14) as DH Group # field and the Key Exchange Data.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD which contains 2 (D-H Group 2) as Notification Data. The message's IKE_SA Responder's SPI value is set to zero.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.6.9: Creating an IKE_SA without a CHILD_SA

Purpose:

To verify that an IKEv2 device can handles a failure of creating a CHILD_SA during the IKE_AUTH exchange.

References:

- [RFC 4718] - Sections 4.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |-------X        | IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| INFORMATIONAL request (HDR, SK {})
     |                | (Packet #3)
     |--------------->| INFORMATIONAL response (HDR, SK {})
     |                | (Judgment #3)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #17

Packet #2: IKE_AUTH requet

Packet #2 is same as Common Packet #3 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request with unacceptable SA proposal for the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT never transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- Step 4
  The NUT can transmits an IKE_AUTH response with a Notify payload of type NO_PROPOSAL_CHOSEN.

Group 1.7. Traffic Selector Negotiation

Test IKEv2.EN.R.1.1.7.1: Narrowing Traffic Selectors

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1	TCP	ANY	NUT	TCP	ANY
Outbound	NUT	TCP	ANY	TN1	TCP	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)      (End-Node)
     |                |
     |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                |  (Packet #1)
     |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                |  (Judgment #1)
     |                |
     |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                |  (Packet #2)
     |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                |  (Judgment #2)
     |                |
     |<---------------|  IPsec {TCP SYN}
     |                |  (Packet #3)
     |--------------->|  IPsec {TCP RST}
     |                |  (Judgment #3)
     |                |
     |<---------------|  IPsec {ICMPv6 Echo Request}
     |                |  (Packet #4)
     |-------X        |  IPsec {ICMPv6 Echo Reply}
     |                |  (Judgment #4)
     |                |
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535

		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- Packet #3: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The cryptographic checksum of the entire message
TCP Header	Source Port	500
	Destination Port	500
	Flags	SYN (0x02)

Part A (BASIC)

1. TN1 sends an IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 sends an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port on NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. The Traffic Selector is narrowed to allow only TCP (6) as IP Protocol.

Step 6: Judgment #3
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT never transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.7.2: TS_UNACCEPTABLE

Purpose:

To verify an IKEv2 device properly handles the Traffice Selector.

References:

- [RFC 4306] - Sections 2.8 and 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1	TCP	ANY	NUT	TCP	ANY
Outbound	NUT	TCP	ANY	TN1	TCP	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi, TSr})
     |                | (Packet #3)
     |-------X        | CREATE_CHILD_SA response (HDR, SK {N, SA, Nr, KEr, TSi, TSr})
     |                |   or
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N(TS_UNACCEPTABLE)})
     |                | (Judgment #3)
     V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- Packet #3: TCP SYN packet

IPv6 Header	Same as the Common Packet #7	
UDP Header	Same as the Common Packet #7	
IKEv2 Header	Same as the Common Packet #7	
E Payload	Same as the Common Packet #7	
N Payload	Same as the Common Packet #7	
SA Payload	Same as the Common Packet #7	
Ni, Nr Payload	Same as the Common Packet #7	
TSi Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A

		Ending Address	NUT's Global Address on Link A

Part A (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request including ICMPv6 (58) as IP Protocol ID value in Traffic Selector Payload to create new CHILD_SA.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits a CREATE_CHILD_SA response or transmits a CREATE_CHILD_SA response including a Notify payload of type TS_UNACCEPTABLE.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.7.3: Narrowing Traffic Selectors from multiple Traffic Selector

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.8
- [RFC4718] - Section 4.10

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1	TCP	ANY	NUT	TCP	ANY
Outbound	NUT	TCP	ANY	TN1	TCP	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)      (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| IPsec {TCP SYN}
    |                | (Packet #3)
    |--------------->| IPsec {TCP RST}
    |                | (Judgment #3)
    |                |
    |<---------------| IPsec {ICMPv6 Echo Request}
    |                | (Packet #4)
    |-------X        | IPsec {ICMPv6 Echo Reply}
    |                | (Judgment #4)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X

		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- Packet #3: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The cryptographic checksum of the entire message
TCP Header	Source Port	500
	Destination Port	500
	Flags	SYN (0x02)

Part A (BASIC)

1. TN1 sends an IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 sends an IKE_AUTH request to the NUT. The message includes two Traffice Selectors. One is set to 6 (TCP) as IP Protocol. Another is set to 58 (IPV6-ICMP).
4. Observe the messages transmitted on Link A.
5. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port on NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. The Traffic Selector Payload has one Traffic Selector with IP Protocol 6 (TCP) to narrow the proposed Traffic Selectors.

Step 6: Judgment #3
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT never transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.8. Error Handling

Test IKEv2.EN.R.1.1.8.1: INVALID_IKE_SPI

Purpose:

To verify an IKEv2 device properly handles IKE messages outside the context of IKE_SA.

References:

- [RFC 4306] - Sections 2.21

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr2, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |-------X        | CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                |   or
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(INVALID_IKE_SPI)})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request (Part A)

IPv6 Header	Same as the Common Packet #13

UDP Header		Same as the Common Packet #13
IKEv2 Header		Other fields are same as the Common Packet #13
	IKE_SA Initiator's SPI	The IKE_SA Initiator' s SPI value used by this IKE message plus 1
	IKE_SA Responder's SPI	The IKE_SA Responder' s SPI value used by this IKE message
E Payload		Same as the Common Packet #13
N Payload		Same as the Common Packet #13
N Payload		Same as the Common Packet #13
SA Payload		Same as the Common Packet #13
Ni, Nr Payload		Same as the Common Packet #13
TSi Payload		Same as the Common Packet #13
TSr Payload		Same as the Common Packet #13

- Packet #3: CREATE_CHILD_SA request (Part A)

IPv6 Header		Same as the Common Packet #13
UDP Header		Same as the Common Packet #13
IKEv2 Header		Other fields are same as the Common Packet #13
	IKE_SA Initiator's SPI	The IKE_SA Initiator' s SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder' s SPI value used by this IKE message plus 1
E Payload		Same as the Common Packet #13
N Payload		Same as the Common Packet #13
N Payload		Same as the Common Packet #13
SA Payload		Same as the Common Packet #13
Ni, Nr Payload		Same as the Common Packet #13
TSi Payload		Same as the Common Packet #13
TSr Payload		Same as the Common Packet #13

Part A: Different IKE_SA Initiator's SPI (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which contains different IKE_SA Initiator's SPI value from IKE_SA Initiator's SPI value in the IKE_AUTH request in Step 3.
6. Observe the messages transmitted on Link A.

Part B: Different IKE_SA Responder's SPI (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which contains different IKE_SA Responder's SPI value from IKE_SA Responder's SPI value in the IKE_AUTH request in Step 4.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits any packets or may transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_IKE_SPI.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT does not transmits any packets or may transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_IKE_SPI.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.8.2: INVALID_SYNTAX

Purpose:

To verify an IKEv2 device properly handles IKE message with an invalid syntax.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr2, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
    |                | (Packet #3)
    |-------X        | CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, KEr, TSi, TSr})
    |                |   or
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(INVALID_SYNTAX)})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #13
UDP Header	Same as the Common Packet #13
IKEv2 Header	Same as the Common Packet #13
E Payload	Same as the Common Packet #13
N Payload	Same as the Common Packet #13

N Payload	Same as the Common Packet #13	
SA Payload	Same as the Common Packet #13	
Ni, Nr Payload	Other fields are same as the common packet #13	
	Payload Length	4
	Nonce Data	empty
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which has no data as Nonce Data as Ni payload.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits a CREATE_CHILD_SA response or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_SYNTAX.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.8.3: INVALID_SELECTORS

Purpose:

To verify an IKEv2 device properly handles an ESP or AH packet whose selectors do not match those of the CHILD_SA.

References:

- [RFC 4306] - Sections 3.10.1
- [RFC 4307] - Sections 7.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			**Destination**		
	Address Range	**Next Layer Protocol**	**Port Range**	**Address Range**	**Next Layer Protocol**	**Port Range**
Inbound	TN1	TCP	ANY	NUT	TCP	ANY
Outbound	NUT	TCP	ANY	TN1	TCP	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Packet #2)
    |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    |<--------------| IPsec {Echo Request}
    |               | (Packet #3)
    |-------------->| INFORMATIONAL request (HDR, SK {N(INVALID_SELECTORS)}
    |               | (Judgment #3)
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as the above table to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATIONAL request with a Notify of type INVALID_SELECTORS.

Possible Problems:

- None.

Group 1.9 ID payload

Test IKEv2.EN.R.1.1.9.1: Receiving IDi Payload

Purpose:

To verify an IKEv2 device receives IDi payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDi payload which NUT receives from TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
(End-Node)        (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3

UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Other fields are same as the Common Packet #3	
	Payload Length	According to above configuration
	ID Type	According to above configuration
	Identification Data	According to above configuration
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

Part A: ID_IPV4_ADDR (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
4. Observe the messages transmitted on Link A.

Part B: ID_FQDN (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
8. Observe the messages transmitted on Link A.

Part C: ID_RFC822_ADDR (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.9.2: Sending IDr Payload

Purpose:

To verify an IKEv2 device transmits IDr payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDr payload which NUT send to TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3

Part A: ID_IPV4_ADDR (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Part B: ID_FQDN (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
8. Observe the messages transmitted on Link A.

Part C: ID_RFC822_ADDR (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Possible Problems:

- None.

Group 1.10 Certificate payload

Test IKEv2.EN.R.1.1.10.1: Sending Certificate Payload

Purpose:

To verify an IKEv2 device handles a CERTREQ payload and transmits a CERT payload propoerly.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Remote	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH, N, SAi2, TSi, TSr})
    |               | (Packet #2)
    |-------------->| IKE_AUTH response (HDR, SK {IDr, CERT, AUTH, N, SAr2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3

UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Next Payload	38 (CERTREQ)
	Oter fields are same as the Common Packet #3	
CERTREQ Payload	See below	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

CERTREQ Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Authority	any

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request with a CERTREQ payload to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response with a CERT payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding and the NUT’s certificate as Certificate Data.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.10.2: Sending Certificate Request Payload

Purpose:

To verify an IKEv2 device properly transmits CERTREQ payload.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                  TN1
(End-Node)          (End-Node)
    |                    |
    |<-------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                    | (Packet #1)
    |------------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, CERTREQ)
    |                    | (Judgment #1)
    |                    |
    V                    V
```

Packet #1	See Common Packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.10.3: RSA Digital Signature

Purpose:

To verify an IKEv2 device authenticates the corresponding node by RSA Digital Signature.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                TN1
(End-Node)        (End-Node)
    |                 |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, CERTREQ)
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, CERT, AUTH, N, SAi2, TSi, TSr})
    |               | (Packet #2)
    |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    |               |
    |<--------------| IPsec {Echo Request}
    |               | (Packet #3)
    |-------------->| IPsec {Echo Reply}
    |               | (Judgment #3)
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3
UDP Header	Same as the Common Packet #3

IKEv2 Header	Same as the Common Packet #3		
E Payload	Same as the Common Packet #3		
IDi Payload	Next Payload	37 (CERT)	
	Oter fields are same as the Common Packet #3		
CERT Payload	See below		
AUTH Payload	Same as the Common Packet #3		
N Payload	Same as the Common Packet #3		
SA Payload	Same as the Common Packet #3		
TSi Payload	Same as the Common Packet #3		
TSr Payload	Same as the Common Packet #3		

CERT Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Data	any

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request with a CERT payload to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.10.4: HEX string PSK

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key.

References:

- [RFC 4306] - Sections 2.15

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Key Value
Local	0xabadcafeabadcafeabadcafeabadcafe (128 bit binary string)

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |  (Packet #1)
    |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |  (Judgment #1)
    |                |
    |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                |  (Packet #2)
    |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                |  (Judgment #2)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Group 1.11 Invalid Values

Test IKEv2.EN.R.1.1.11.1: Non zero RESERVED fields in IKE_SA_INIT request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     V                V
```

Packet #1	See Common Packet #1 All RESERVED fields are set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.11.2: Non zero RESERVED fields in IKE_AUTH request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT             TN1
 (End-Node)      (End-Node)
     |               |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |               | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |               | (Judgment #1)
     |               |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |               | (Packet #2)
     |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |               | (Judgment #2)
     |               |
     V               V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3 All RESERVED fields are set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.11.3: Version bit is set

Purpose:

To verify an IKEv2 device ignores the content of Version bit in IKE messages.

References:

- [RFC 4306] - Sections 3.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)       (End-Node)
   |                |
   |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                | (Judgment #1)
   |                |
   V                V
```

Packet #1	See Common Packet #1 Version bit is set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request whose Version bit is set to one.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.11.4: Response bit is set

Purpose:

To verify an IKEv2 device ignores an IKE request message whose Response bit is set.

References:

- [RFC 4306] - Sections 2.21

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)        (End-Node)
    |                  |
    |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                  | (Packet #1)
    |-------X          | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                  | (Judgment #1)
    |                  |
    V                  V
```

Packet #1	See Common Packet #1 Response bit is set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request whose Response bit is set to one.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT never responds with an IKE_SA_INIT response to an IKE_SA_INIT request from the TN1.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.11.5: Unrecognized Notify Message Type

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type in IKE messages.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)       (End-Node)
    |                 |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni, N)
    |                 | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                 | (Judgment #1)
    |                 |
    V                 V

N: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #1	
UDP Header	All fields are same as Common Packet #1	
IKEv2 Header	All fields are same as Common Packet #1	
SA Payload	All fields are same as Common Packet #1	
KE Payload	All fields are same as Common Packet #1	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #1	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	See each part description.

Part A: Unrecognized Notify Message Type of error 16383 (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request with a Notify payload of unrecognized Notify Message Type value.

2. Observe the messages transmitted on Link A.

Part B: Unrecognized Notify Message Type of status 65535 (BASIC)

3. TN starts to negotiate with NUT by sending IKE_SA_INIT request with a Notify payload of unrecognized Notify Message Type value.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Possible Problems:

- None.

Group 2. The CREATE_CHILD_SA Exchange

Group 2.1. Header and Payload Formats

Test IKEv2.EN.R.1.2.1.1: Receipt of CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.3 and 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Judgment #3)
    |                |
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #13

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
6. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
12. Observe the messages transmitted on Link A.

Part D: Notify Payload (USE_TRANSPORT_MODE) Format (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
18. Observe the messages transmitted on Link A.

Part E: SA Payload Format (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
24. Observe the messages transmitted on Link A.

Part F: Nonce Payload Format (BASIC)

25. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
28. Observe the messages transmitted on Link A.
29. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
30. Observe the messages transmitted on Link A.

Part G: TSi Payload Format (BASIC)

31. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
32. Observe the messages transmitted on Link A.

33. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
34. Observe the messages transmitted on Link A.
35. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
36. Observe the messages transmitted on Link A.

Part H: TSr Payload Format (BASIC)

37. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
38. Observe the messages transmitted on Link A.
39. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
40. Observe the messages transmitted on Link A.
41. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
42. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 73 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.
- A Next Payload field is set to Encrypted Payload (46).

- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to CREATE_CHILD_SA (36).
- A Flags field is set to (00000100)2 = (4)10.
- A Message ID field is set to the same value as corresponding IKEv2 request message's Message ID.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 8: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 74 Encrypted payload

- A Next Payload field is set to N Payload (41).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire

message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 75 Notify Payload format

- A Next Payload field is set to SA Payload (33).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload. It is 8 bytes for USE_TRANSPORT_MODE.
- A Protocol ID field is set to undefined (0).
- A SPI Size field is set to zero.
- A Notify Message Type field is set to USE_TRANSPORT_MODE (16391)

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3

```
                                1                   2                   3
                0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
               ! Next     44   !0!      0      ! Length          40           !             |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
               !       0       !       0       ! Length          36           !  |          |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
               ! Number     1  ! Prot ID    3  ! SPI Size   4  ! Trans Cnt  3 !  |          |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
               ! SPI value                                                    !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
            |  !       3       !       0       ! Length           8           !  |          |
Transform   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |SA Payload
            |  ! Type  1  (EN) !       0       ! Transform ID     3     (3DES) !  | Proposal |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
            |  !       3       !       0       ! Length           8           !  |          |
Transform   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
            |  ! Type  3  (IN) !       0       ! Transform ID     2     (SHA1) !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
            |  !       0       !       0       ! Length           8           !  |          |
Transform   |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
            |  ! Type  5  (ESN)!       0       ! Transform ID     0       (No)  !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 76 SA Payload contents

The NUT transmits a CREATE_CHILD_SA response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 77 SA Payload format

- A Next Payload field is set to Nr Payload (40).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 78 Proposal sub-structure format

Proposal #1

- A 0 or 2 field is set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field is set to zero.
- A Proposal Length field is set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field is set to 1.
- A Protocol ID field is set to ESP (3).
- A SPI Size field is set to 4.
- A # of Transforms field is set to 3.
- A SPI field is set to the sending entity's SPI (4 octets value)

Transform field is set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 79 Transform sub-structure format

Transform #1

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field is set to ENCR (1).
- A RESERVED field is set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.

- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field is set to INTEG (3).
- A RESERVED field is set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field is set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field is set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field is set to ESN (5).
- A RESERVED field is set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part E

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 30: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 80 Nonce Payload format

- A Next Payload field is set to TSi Payload (44).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Nonce Data field is set to random data generated by the transmitting entity.
- The size of the Nonce must between 16 and 256 octets.

Part F

Step 32: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 34: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 36: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 81 TSi Payload format

- A Next Payload field is set to TSr Payload (45).
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to 1.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 82 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header.

- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to TN1 address.
- A Ending Address field is set to TN1 address.

Part G

Step 38: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted lgorithms.

Step 40: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 42: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 83 TSr Payload format

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length of the current payload.
- A Number of TSs field is set to 1.
- A RESERVED field is set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 84 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field is set to zero.
- A Selector Length field is set to length of this Traffic Selector Substructure including the header.
- A Start Port field is set to zero.
- An End Port field is set to 65535.
- A Starting Address field is set to NUT address.
- An Ending Address field is set to NUT address.

Possible Problems:

- CREATE_CHILD_SA response has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, Nr, [KEr], TSi, TSr,
[N(ADDITIONAL_TS_POSSIBLE)]
```

- Each of transforms can be located in the any order.

Group 2.2. Use of Retransmission Timers

Test IKEv2.EN.R.1.2.2.1: Receipt of retransmitted CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device retransmits CREATE_CHILD_SA request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Packet #3)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N, N+, SA, Nr, TSi, TSr})
     |                | (Judgment #3)
     |                |
     |                * wait until retrans timer expires
     |-------X        | CREATE_CHILD_SA response (HDR, SK {N, N+, SA, Nr, TSi, TSr})
     |                | (Judgment #4)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Packet #4)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N, N+, SA, Nr, TSi, TSr})
     |                | (Judgment #5)
     |                |
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #13
Packet #4	See Common Packet #13

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 trasmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request to rekey the established CHILD_SAs to the NUT.
6. Observe the messages transmitted on Link A.
7. Observe the messages transmitted on Link A.
8. TN1 retransmits the same message as a CREATE_CHILD_SA request transmitted in Step 5 to the NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #4
The NUT never retransmits a CREATE_CHILD_SA response which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Step 9: Judgment #5
The NUT retransmits a CREATE_CHILD_SA response which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Possible Problems:

- none

Group 2.3. State Synchronization and Connection Timeouts

Test IKEv2.EN.R.1.2.3.1: Receiving Delete Payload for Multiple CHILD_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT             TN1
 (End-Node)      (End-Node)
     |               |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |               | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |               | (Judgment #1)
     |               |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |               | (Packet #2)
     |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |               | (Judgment #2)
     |               |
     |<--------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
     |               | (Packet #3)
     |-------------->| CREATE_CHILD_SA response (HDR, SK {N, SA, Nr, TSi, TSr})
     |               | (Judgment #3)
     |               |
     |<--------------| INFORMATIONAL request (HDR, SK {D})
     |               | (Packet #4)
     |-------------->| INFORMATIONAL request (HDR, SK {D})
     |               | (Judgment #4)
     |               |
     V               V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common below
Packet #3	See Common below
Packet #4	See Common below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Same as the Common Packet #3	
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #7	
UDP Header	Same as the Common Packet #7	
IKEv2 Header	Same as the Common Packet #7	
E Payload	Same as the Common Packet #7	
N Payload	Same as the Common Packet #7	
SA Payload	Same as the Common Packet #7	
Ni, Nr Payload	Same as the Common Packet #7	
TSi Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A

		Ending Address	NUT's Global Address on Link A

- Packet #4: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	16
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	2
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange
		SPI negotiated by CREATE_CHILD_SA exchange

Part A: (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request to establish a new CHILD_SA to the NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an INFORMATIONAL request with a Delete payload including the first negotiated CHILD_SA's inbound SPI and the second negotiated CHILD_SA's inbound SPI.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 8: Judgment #4
The NUT transmits an INFORMATIONAL response with delete payload for SPIs which are negotiated by Initial Exchange and CREATE_CHILD_SA exchange.

Possible Problems:

- INFORMATIONAL response from NUT may not contain Delete Payload by implementation policy. This behavior is defined at section 1.4 in RFC 4306 as an exception.

Group 2.4. Cryptographic Algorithm Negotiation

Test IKEv2.EN.R.1.2.4.1: Sending NO_PROPOSAL_CHOSEN

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA request with an unacceptable SA payload.

References:

- [RFC 4306] - Sections 2.7 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Packet #3)
     |-------X        | CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
     |                |   or
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
     |                | (Judgment #3)
     |                |
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #13	
UDP Header	Same as the Common Packet #13	
IKEv2 Header	Same as the Common Packet #13	
E Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
SA Payload	Other fields are same as the Common Packet #13	
	SA Proposals	See below
Ni, Nr Payload	Same as the Common Packet #13	
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		36
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		3
		SPI		any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	12 (AES_CBC)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	5 (AES_XCBC_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	1 (ESN)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 trasmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request to rekey the established CHILD_SAs to the NUT. The CREATE_CHILD_SA request includes a SA payload with a proposal unaccepted by the NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmit a CREATE_CHILD_SA response or transmits a CREATE_CHILD_SA response including a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- None.

Group 2.5. Rekeying CHILD_SA Using a CREATE_CHILD_SA exchange

Test IKEv2.EN.R.1.2.5.1: Close the replaced CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to rekey CHILD_SA and INFORMATIONAL Excahnges to delete old CHILD_SAs.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| IPsec {Echo Request}
    |                | (Packet #3)
    |--------------->| IPsec {Echo Reply}
    |                | (Judgment #3)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #4)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Judgment #4)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {D})
    |                | (Packet #5)
    |--------------->| INFORMATIONAL response (HDR, SK {D})
    |                | (Judgment #5)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19
Packet #4	See Common Packet #13
Packet #5	See below

- Packet #5: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request including a Delete payload with the old CHILD_SA's SPI value to the NUT.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4

The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 10: Judgment #5
The NUT transmits an INFORMATIONAL response including a Delete payload with the old CHILD_SA’s SPI value to the TN1.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.5.2: Use of the new CHILD_SA

Purpose:

To verify an IKEv2 device properly handle old CHILD_SA and new CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)      (End-Node)
    |                |
    |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |  (Packet #1)
    |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |  (Judgment #1)
    |                |
    |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                |  (Packet #2)
    |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                |  (Judgment #2)
    |                |
    |<---------------|  IPsec {Echo Request}
    |                |  (Packet #3)
    |--------------->|  IPsec {Echo Reply}
    |                |  (Judgment #3)
    |                |
    |<---------------|  CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                |  (Packet #4)
    |--------------->|  CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                |  (Judgment #4)
    |                |
    |<---------------|  INFORMATIONAL request (HDR, SK {D})
    |                |  (Packet #5)
    |--------------->|  INFORMATIONAL response (HDR, SK {D})
    |                |  (Judgment #5)
    |                |
    |<---------------|  IPsec {Echo Request} (new SA)
    |                |  (Packet #6)
    |--------------->|  IPsec {Echo Reply} (new SA)
    |                |  (Judgment #6)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19 (CHILD_SA is negotiated by steps 1 through 4.)
Packet #4	See Common Packet #13
Packet #5	See below
Packet #6	See Common Packet #19 (CHILD_SA is negotiated by steps 7 through 8.)

- Packet #5: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request including a Delete payload with the old CHILD_SA's SPI value to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 10: Judgment #5
The NUT transmits an INFORMATIONAL response including a Delete payload with the old CHILD_SA’s SPI value to the TN1.

Step 12: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using the newly negotiated algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.5.3: Receiving Multiple Transform

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)       (End-Node)
    |                 |
    |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Packet #1)
    |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                 | (Judgment #1)
    |                 |
    |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 | (Packet #2)
    |---------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 | (Judgment #2)
    |                 |
    |<----------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                 | (Packet #3)
    |---------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                 | (Judgment #3)
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

From part A to part C, TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the transforms as follows:

	CREATE_CHILD_SA exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN

Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #13	
UDP Header	Same as the Common Packet #13	
IKEv2 Header	Same as the Common Packet #13	
E Payload	Same as the Common Packet #13	
IDi Payload	Same as the Common Packet #13	
AUTH Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
SA Payload	Other fields are same as the Common Packet #13	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part C

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.5.4: Receiving Multiple Proposal

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the two proposals as follows:

	CREATE_CHILD_SA exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN

Part B	Proposal #1	ESP	ENCR_3DES	**AUTH_AES_XCBC_96**	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part C	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	**ESN**
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #13	
UDP Header	Same as the Common Packet #13	
IKEv2 Header	Same as the Common Packet #13	
E Payload	Same as the Common Packet #13	
IDi Payload	Same as the Common Packet #13	
AUTH Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
SA Payload	Other fields are same as the Common Packet #13	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		2
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)

			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.5.5: Perfect Forward Secrecy

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA exchange when Perfect Forward Secrecy enabels.

References:

- [RFC 4306] - Sections 2.12

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. Enable PFS.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT               TN1
(End-Node)        (End-Node)
    |                 |
    |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                 | (Packet #1)
    |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                 | (Judgment #1)
    |                 |
    |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                 | (Packet #2)
    |---------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                 | (Judgment #2)
    |                 |
    |<----------------| IPsec {Echo Request}
    |                 | (Packet #3)
    |---------------->| IPsec {Echo Reply}
    |                 | (Judgment #3)
    |                 |
    |<----------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, KEi, TSi, TSr})
    |                 | (Packet #4)
    |---------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, KEr, TSi, TSr})
    |                 | (Judgment #4)
    |                 |
    |<----------------| INFORMATIONAL request (HDR, SK {D})
    |                 | (Packet #5)
    |---------------->| INFORMATIONAL response (HDR, SK {D})
    |                 | (Judgment #5)
    |                 |
    |<----------------| IPsec {Echo Request} (new SA)
    |                 | (Packet #6)
    |---------------->| IPsec {Echo Reply} (new SA)
    |                 | (Judgment #6)
    V                 V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19 (CHILD_SA is negotiated by steps 1 through 4.)
Packet #4	See below
Packet #5	See below
Packet #6	See Common Packet #19 (CHILD_SA is negotiated by steps 7 through 8.)

- Packet #4: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #13	
UDP Header	Same as the Common Packet #13	
IKEv2 Header	Same as the Common Packet #13	
E Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
N Payload	Same as the Common Packet #13	
SA Payload	Same as the Common Packet #13	
Ni Payload	Next Payload	34 (KE)
KEi Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	136

	DH Group #	2
	Reserved	0
	Key Exchange Data	any
TSi Payload	Same as the Common Packet #13	
TSr Payload	Same as the Common Packet #13	

- Packet #5: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request including a Delete payload with the old CHILD_SA's SPI value to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 10: Judgment #5
The NUT transmits an INFORMATIONAL response including a Delete payload with the old CHILD_SA’s SPI value to the TN1.

Step 12: Judgment #6
The NUT transmits an Echo Reply with IPsec ESP using the newly negotiated algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.5.6: Use of the old CHILD_SA

Purpose:

To verify an IKEv2 device properly handle old CHILD_SA and new CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #3)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #3)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
     |                | (Packet #4)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
     |                | (Judgment #4)
     |                |
     |<---------------| IPsec {Echo Request} (old CHILD_SA)
     |                | (Packet #5)
     |--------------->| IPsec {Echo Reply} (old CHILD_SA)
     |                | (Judgment #5)
     V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19 (CHILD_SA is negotiated by steps 1 through 4.)

Packet #4	See Common Packet #13
Packet #5	See Common Packet #19 (CHILD_SA is negotiated by steps 1 through 4.)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms again.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 10: Judgment #5
The NUT transmits an Echo Reply with IPsec ESP using the first negotiated algorithms.

Possible Problems:

- none

Group 2.6. Rekeying IKE_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.EN.R.1.2.6.1: Sending CREATE_CHILD_SA response

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA Excahnge to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8 and 2.18

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     |<---------------| IPsec {Echo Request}
     |                | (Packet #3)
     |--------------->| IPsec {Echo Reply}
     |                | (Judgment #3)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                | (Packet #4)
     |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
     |                | (Judgment #4)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19

Packet #4	See Common Packet #11

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA's initiator's SPI value.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's responder's SPI value in the SPI field.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.EN.R.1.2.6.2: Receipt of cryptographically valid message on the old SA

Purpose:

To verify an IKEv2 device properly uses old IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                  TN1
(End-Node)          (End-Node)
    |                    |
    |<-------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                    | (Packet #1)
    |------------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                    | (Judgment #1)
    |                    |
    |<-------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                    | (Packet #2)
    |------------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                    | (Judgment #2)
    |                    |
    |<-------------------| IPsec {Echo Request}
    |                    | (Packet #3)
    |------------------->| IPsec {Echo Reply}
    |                    | (Judgment #3)
    |                    |
    |<-------------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                    | (Packet #4)
    |------------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |                    | (Judgment #4)
    |                    |
    |<-------------------| INFORMATIONAL request (HDR, SK {}) (old IKE_SA)
    |                    | (Packet #5)
    |------------------->| INFORMATIONAL response (HDR, SK {}) (old IKE_SA)
    |                    | (Judgment #5)
    V                    V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19

Packet #4	See Common Packet #11
Packet #5	See Common Packet #17 (CHILD_SA is negotiated by steps 1 through 4.)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request with no payloads protected by the old IKE_SA.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s responder’s SPI value in the SPI field.

Step 10: Judgment #5
The NUT responds with an INFORMATIONAL response with no payloads protected by the old IKE_SA.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.6.3: Receipt of cryptographically valid message on the new SA

Purpose:

To verify an IKEv2 device properly uses new IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT               TN1
(End-Node)        (End-Node)
     |                 |
     |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                 | (Packet #1)
     |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                 | (Judgment #1)
     |                 |
     |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                 | (Packet #2)
     |---------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                 | (Judgment #2)
     |                 |
     |<----------------| IPsec {Echo Request}
     |                 | (Packet #3)
     |---------------->| IPsec {Echo Reply}
     |                 | (Judgment #3)
     |                 |
     |<----------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |                 | (Packet #4)
     |---------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
     |                 | (Judgment #4)
     |                 |
     |<----------------| INFORMATIONAL request (HDR, SK {})
     |                 | (Packet #5)
     |---------------->| INFORMATIONAL response (HDR, SK {})
     |                 | (Judgment #5)
     V                 V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19

Packet #4	See Common Packet #11
Packet #5	See Common Packet #17

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's responder's SPI value in the SPI field.

Step 10: Judgment #5
The NUT responds with an INFORMATIONAL response with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.6.4: Close the replaced IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.8 and 5.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT            TN1
(End-Node)     (End-Node)
    |              |
    |<-------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |              | (Packet #1)
    |------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |              | (Judgment #1)
    |              |
    |<-------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |              | (Packet #2)
    |------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |              | (Judgment #2)
    |              |
    |<-------------| IPsec {Echo Request}
    |              | (Packet #3)
    |------------->| IPsec {Echo Reply}
    |              | (Judgment #3)
    |              |
    |<-------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |              | (Packet #4)
    |------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |              | (Judgment #4)
    |              |
    |<-------------| INFORMATIONAL request (HDR, SK {D})
    |              | (Packet #5)
    |------------->| INFORMATIONAL response (HDR, SK {})
    |              | (Judgment #5)
    |              |
    |<-------------| IPsec {Echo Request}
    |              | (Packet #6)
    |------------->| IPsec {Echo Reply}
    |              | (Judgment #6)
    V              V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #19
Packet #4	See Common Packet #11
Packet #5	See below
Packet #6	See Common Packet #19

- Packet #5: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	16
	Procotol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index(es) (SPI)	empty

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA to the NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits an INFORMATIONAL request with a Delete payload which has 1 (IKE_SA) in the Protocol ID field, zero in the SPI Size field and zero in the # of SPIs field.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s responder’s SPI value in the SPI field.

Step 10: Judgment #5
The NUT responds with an INFORMATIONAL response with no payloads.

Step 12: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.

Possible Problems:

- none.

Test IKEv2.EN.R.1.2.6.5: Receiving Multiple Transform

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transform to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                TN1
(End-Node)         (End-Node)
    |                  |
    |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                  | (Packet #1)
    |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                  | (Judgment #1)
    |                  |
    |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                  | (Packet #2)
    |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                  | (Judgment #2)
    |                  |
    |<-----------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                  | (Packet #3)
    |----------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr })
    |                  | (Judgment #3)
    V                  V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the transforms as follows:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_AES_CBC** **ENCR_3DES**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_AES128_CBC** **PRF_HMAC_SHA1**	AUTH_HMAC_SHA1_96	Group 2

Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96** **AUTH_HMAC_SHA1_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14** **Group 2**

- Packet #3 CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #11	
UDP Header	Same as the Common Packet #11	
IKEv2 Header	Same as the Common Packet #11	
SA Payload	Other fields are same as the common packet #11	
	SA Proposals	See SA Table below
Ni, Nr Payload	Same as the Common Packet #11	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

6. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo Random Function (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithm (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Multiple D-H Group (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2"as proposed algorithms.

Part B

Step 8: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part D

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.6.6: Receiving Multiple Proposal

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple proposal to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT             TN1
(End-Node)      (End-Node)
     |               |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |               | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |               | (Judgment #1)
     |               |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |               | (Packet #2)
     |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |               | (Judgment #2)
     |               |
     |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
     |               | (Packet #3)
     |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
     |               | (Judgment #3)
     V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See below

TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the two proposals as follows:

	IKE_SA_INIT exchanges Algorithms					
	Proposals	Protocol ID	Encryption	PRF	Integrity	D-H Group
Part A	Proposal #1	IKE	ENCR_AES_CBC	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

Part B	Proposal #1	IKE	ENCR_3DES	**PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96**	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part D	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14**
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #11	
UDP Header	Same as the Common Packet #11	
IKEv2 Header	Same as the Common Packet #11	
SA Payload	Other fields are same as the common packet #11	
	SA Proposals	See SA Table below
Ni, Nr Payload	Same as the Common Packet #11	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		2
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)

		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo Rnadom Function (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Multiple D-H Group (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type

REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.6.7: Changing PRFs when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly uses new IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  Configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA Rekeying Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |               | (Packet #2)
    |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |               | (Packet #3)
    |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
    |               | (Judgment #3)
    |               |
    |<--------------| INFORMATIONAL request (HDR, SK {})
    |               | (Packet #4)
    |-------------->| INFORMATIONAL response (HDR, SK {})
    |               | (Judgment #4)
    V               V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1

Packet #2	See Common Packet #3
Packet #3	Seebelow
Packet #4	See Common Packet #17

Packet #3: CREATE_CHILD_SA request

Packet #3 is same as Common Packet #11 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	14 (2048 MODP Group)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request to the NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an INFORMATIONAL request with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s responder’s SPI value in the SPI field.

Step 8: Judgment #4

The NUT responds with an INFORMATIONAL response with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.

Possible Problems:

- none

Test IKEv2.EN.R.1.2.6.8: D-H transform NONE when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly handles D-H transform NONE when rekeying the IKE_SA.

References:

- [RFC 4718] - Sections 5.12

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
    |                | (Packet #3)
    |-------X        | CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	Seebelow

Packet #3: CREATE_CHILD_SA request

Packet #3 is same as Common Packet #11 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	0 (NONE)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value. The message proposes D-H transform NONE.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- none

Group 2.7. Creating new CHILD_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.EN.R.1.2.7.1: Receipt of cryptographically valid message on the new SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to create a new CHILD_SA.

References:

- [RFC 4306] - Sections 2.8 and 2.18

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(End-Node)       (End-Node)
   |                |
   |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                |  (Packet #1)
   |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                |  (Judgment #1)
   |                |
   |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
   |                |  (Packet #2)
   |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
   |                |  (Judgment #2)
   |                |
   |<---------------|  IPsec {TCP-SYN}
   |                |  (Packet #3)
   |--------------->|  IPsec {TCP-RST}
   |                |  (Judgment #3)
   |                |
   |<---------------|  IPsec {Echo Request}
   |                |  (Packet #4)
   |-------X        |  IPsec {Echo Reply}
   |                |  (Judgment #4)
   |                |
   |<---------------|  CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                |  (Packet #5)
   |--------------->|  CREATE_CHILD_SA response (HDR, SK {N, SA, Nr, TSi, TSr})
   |                |  (Judgment #5)
   |                |
   |<---------------|  IPsec {TCP-SYN}
   |                |  (Packet #6)
   |--------------->|  IPsec {TCP-RST}
   |                |  (Judgment #6)
   |                |
   |<---------------|  IPsec {Echo Request}
   |                |  (Packet #7)
   |--------------->|  IPsec {Echo Reply}
   |                |  (Judgment #7)
   |                |
   V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #19
Packet #5	See below
Packet #6	See below
Packet #7	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #3
UDP Header	Same as the Common Packet #3
IKEv2 Header	Same as the Common Packet #3
E Payload	Same as the Common Packet #3
IDi Payload	Same as the Common Packet #3
AUTH Payload	Same as the Common Packet #3
N Payload	Same as the Common Packet #3
SA Payload	Same as the Common Packet #3

TSi Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #3	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link A
		Ending Address	TN1's Global Address on Link A

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link X
		Ending Address	NUT's Global Address on Link X

- Packet #3: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

- Packet #5: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #7	
UDP Header	Same as the Common Packet #7	
IKEv2 Header	Same as the Common Packet #7	
E Payload	Same as the Common Packet #7	
IDi Payload	Same as the Common Packet #7	
AUTH Payload	Same as the Common Packet #7	
N Payload	Same as the Common Packet #7	
SA Payload	Same as the Common Packet #7	
TSi Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #7	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TN1's Global Address on Link X
		Ending Address	TN1's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)

		IP Protocol ID	58 (IPV6-ICMP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	NUT's Global Address on Link A
		Ending Address	NUT's Global Address on Link A

- Packet #6: TCP SYN packet

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port 30000 on NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a TCP-SYN packet with IPsec ESP using corresponding algorithms to closed port 30000 on NUT.
12. Observe the messages transmitted on Link A.
13. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 8: Judgment #4
The NUT never transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 12: Judgment #6
The NUT transmits a TCP-RST packet with IPsec ESP using corresponding algorithms.

Step 14: Judgment #7
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- If the NUT uses TCP port 30000 for other applications, the TN1 transmits TCP-SYN packets to other closed TCP port on the NUT.

Group 2.8. Error Handling

Test IKEv2.EN.R.1.2.8.1: AUTHENTICATION_FAILED

Purpose:

To verify an IKEv2 device properly handles AUTHENTICATION_FAILED message.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)       (End-Node)
     |                |
     |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
     |                | (Packet #2)
     |-------X        | IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
     |                |   or
     |-------------->| IKE_AUTH response (HDR, N(AUTHENTICATION_FAILED))
     |                | (Judgment #2)
     |                |
     V                V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2 (Part A): IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Other fields are same as the Common Packet #3	
	Payload Length	8

	Auth Method	2 (SK_MIC)
	Authentication Data	empty
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

- Packet #2 (Part B): IKE_AUTH request

IPv6 Header	Same as the Common Packet #3	
UDP Header	Same as the Common Packet #3	
IKEv2 Header	Same as the Common Packet #3	
E Payload	Same as the Common Packet #3	
IDi Payload	Same as the Common Packet #3	
AUTH Payload	Other fields are same as the Common Packet #3	
	Payload Length	28
	Auth Method	1 (RSA_DS)
	Authentication Data	Same data as the common packet #3 (caluculated by using SK_MIC)
N Payload	Same as the Common Packet #3	
SA Payload	Same as the Common Packet #3	
TSi Payload	Same as the Common Packet #3	
TSr Payload	Same as the Common Packet #3	

Part A Invalid Authentication Data (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request which has an invalid Authentication Data in AUTH payload to the NUT.
4. Observe the messages transmitted on Link A.

Part B Invalid Auth method (ADVANCED)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request which has an invalid Auth Method in AUTH payload to the NUT.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT does not transmit an IKE_AUTH response or transmits an IKE_AUTH response with Notify payload of type AUTHENTICATION_FAILED without encryption to the TN1.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT does not transmit an IKE_AUTH response or transmits an IKE_AUTH response with Notify payload of type AUTHENTICATION_FAILED without encryption to the TN1.

Possible Problems:

- None.

Group 2.9. Non zero RESERVED fields

Test IKEv2.EN.R.1.2.9.1: Non zero RESERVED fields in CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, N+, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N+, SA, Nr, TSi, TSr})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #13 All RESERVED fields are set to one.

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- none

Group 3. The INFORMATIONAL Exchange

Group 3.1. Header and Payload Formats

Test IKEv2.EN.R.1.3.1.1: Sending INFORMATIONAL response

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.1.2, 1.4, 3.1 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
     NUT                TN1
 (End-Node)         (End-Node)
      |                  |
      |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
      |                  | (Packet #1)
      |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
      |                  | (Judgment #1)
      |                  |
      |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
      |                  | (Packet #2)
      |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
      |                  | (Judgment #2)
      |                  |
      |<-----------------| INFORMATIONAL request (HDR, SK { })
      |                  | (Packet #3)
      |----------------->| INFORMATIONAL response (HDR, SK { })
      |                  | (Judgment #3)
      |                  |
      V                  V

 N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #17

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT_SA Response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT_SA Response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 85 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.

- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.
- An Exchange Type field is set to INFORMATIONAL (37).
- A Flags field is set to (00000100)2 = (4)10.
- A Message ID field is set to the same value as corresponding IKEv2 request message's Message ID.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 9: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 11: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 14: Judgment #3

The NUT transmits an INFORMATIONAL response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 86 Encrypted payload

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.

- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Possible Problems:

- None.

Group 3.2. Use of Retransmission Timers

Test IKEv2.EN.R.1.3.2.1: Receipt of retransmitted INFORMATIONAL request

Purpose:

To verify an IKEv2 device properly handles the retransmission.

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
(End-Node)       (End-Node)
    |                |
    |<---------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |  (Packet #1)
    |--------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |  (Judgment #1)
    |                |
    |<---------------|  IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                |  (Packet #2)
    |--------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                |  (Judgment #2)
    |                |
    |<---------------|  INFORMATIONAL request (HDR, SK { })
    |                |  (Packet #3)
    |--------------->|  INFORMATIONAL response (HDR, SK { })
    |                |  (Judgment #3)
    |                |
    |                *  wait until retrans timer expires
    |--------------->|  INFORMATIONAL response (HDR, SK { })
    |                |  (Judgment #4)
    |                |
    |<---------------|  INFORMATIONAL request (HDR, SK { })
    |                |  (Packet #4)
    |--------------->|  INFORMATIONAL response (HDR, SK { })
    |                |  (Judgment #5)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1

Packet #2	See Common Packet #3
Packet #3	See Common Packet #17
Packet #4	See Common Packet #17 (same Message ID as packet #3)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request with no payloads. The Message ID is the same as step 5.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Step 7: Judgment #4
The NUT never retransmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #5
The NUT transmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- none

Group 3.3. Non zero RESERVED fields

Test IKEv2.EN.R.1.3.3.1: Non RESERVED fields in INFORMATIONAL request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
(End-Node)      (End-Node)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| INFORMATIONAL request (HDR, SK {})
    |                | (Packet #3)
    |--------------->| INFORMATIONAL response (HDR, SK {})
    |                | (Judgment #3)
    |                |
    V                V

N: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #17 All RESERVED fields are set to one.

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads. All RESERVED fields in the message are set to one.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Section 1.2.2. Endpoint to Security Gateway Tunnel

Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.EN.R.2.1.1.1: Sending IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
 (End-Node)         (SGW)
     |                |
     |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Packet #1)
     |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
     |                | (Judgment #1)
     |                |
     |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
     |                | (Packet #2)
     |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
     |                | (Judgment #2)
     |                |
     V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A: IKE Header Format (BASIC)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to NUT.
4. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

5. TN1 transmits an IKE_SA_INIT request to NUT.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an IKE_SA_INIT request to NUT.
8. Observe the messages transmitted on Link A.

Part C: IDr Payload Format (BASIC)

9. TN1 transmits an IKE_SA_INIT request to NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an IKE_SA_INIT request to NUT.
12. Observe the messages transmitted on Link A.

Part D: AUTH Payload Format (BASIC)

13. TN1 transmits an IKE_SA_INIT request to NUT.
14. Observe the messages transmitted on Link A.
15. TN1 transmits an IKE_SA_INIT request to NUT.
16. Observe the messages transmitted on Link A.

Part E: SA Payload Format (BASIC)

17. TN1 transmits an IKE_SA_INIT request to NUT.
18. Observe the messages transmitted on Link A.
19. TN1 transmits an IKE_SA_INIT request to NUT.
20. Observe the messages transmitted on Link A.

Part F: TSi Payload Format (BASIC)

21. TN1 transmits an IKE_SA_INIT request to NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits an IKE_SA_INIT request to NUT.
24. Observe the messages transmitted on Link A.

Part G: TSr Payload Format (BASIC)

25. TN1 transmits an IKE_SA_INIT request to NUT.
26. Observe the messages transmitted on Link A.
27. TN1 transmits an IKE_SA_INIT request to NUT.
28. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 87 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_AUTH (35).
- A Flags field set to (00010000)2 = (16)10.
- A Message ID field set to 1.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 88 Encrypted payload

- A Next Payload field set to IDr Payload (36).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm.
- An Encrypted IKE Payloads field set to encrypted IKE Payloads
- A Padding field set to any value which to be a multiple of the encryption block size.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. The checksum must be valid.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 89 ID Payload format

- A Next Payload field set to AUTH Payload (39).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- An ID Type field set to ID_IPV6_ADDR (5).
- A RESERVED field set to zero.
- An Identification Data field set to the NUT address.

Part D

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 90 AUTH Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- An Auth Method field set to Shared Key Message Integrity Code (2).
- A RESERVED field set to zero.
- An Authentication Data field set to correct authentication value.

Part E

Step 18: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2

```
                                 1                   2                   3
             0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
            ! Next     44    !0!      0      ! Length          40           !              |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
            !       0       !       0       ! Length          36           !  |           |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
            ! Number     1  ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3  !  |           |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
            ! SPI value                                                     !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3       !       0       ! Length          8             !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |SA Payload
          | ! Type  1  (EN) !       0       ! Transform ID     3     (3DES) !  | Proposal  |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3       !       0       ! Length          8             !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  3  (IN) !       0       ! Transform ID     2     (SHA1) !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       0       !       0       ! Length          8             !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  5  (ESN)!       0       ! Transform ID     0       (No) !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 91 SA Payload contents

The NUT transmits an IKE_AUTH response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 92 SA Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

A Proposals field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 93 Proposal sub-structure format

- A 0 or 2 field set to zero (last).
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes.
- A Proposal # field set to 1.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 94 Transform sub-structure format

- A 0 or 3 field set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

- A 0 or 3 field set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

- A 0 or 3 field set to zero.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 24: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 95 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 96 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to NUT address.
- A Ending Address field set to NUT address.

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 97 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 98 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to TN1 address.
- An Ending Address field set to TN1 address.

Possible Problems:

- None.

Test IKEv2.EN.R.2.1.1.2: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1              TH1
(End-Node)         (SGW)           (Host)
    |                |                |
    |<---------------|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                |                | (Packet #1)
    |--------------->|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                |                | (Judgment #1)
    |                |                |
    |<---------------|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                |                | (Packet #2)
    |--------------->|                | IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |                |                | (Judgment #2)
    |                |                |
    |<===============+----------------| IPsec {Echo Request}
    |                |                | (Packet #3)
    |================+--------------->| IPsec {Echo Reply}
    |                |                | (Judgment #3)
    |                |                |
    v                v                v
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #20

Part A (BASIC)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT response to NUT.
4. Observe the messages transmitted on Link A.
5. TH1 transmits an Echo Request and TN1 forwards an Echo Request with IPsec ESP using corresponding algorithms to NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Section 2. Security Gateway
Section 2.1. Initiator
Section 2.1.1. Security Gateway to Security Gateway Tunnel
Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.SGW.I.1.1.1.1: Sending IKE_SA_INIT request

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT request using properly Header and Payloads format

References:

- [RFC4306] - Section 1.2, 2.10, 3.1, 3.2, 3.3, 3.4 and 3.9
- [RFC 4718] - Sections 7.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |                |
  V                V
```

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Part B: SA Payload Format (BASIC)

3. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
4. Observe the messages transmitted on Link A.

Part C: KE Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.

Part D: Nonce Payload Format (BASIC)

7. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 99 Header format

- An IKE_SA Initiator’s SPI field set to a 64-bits value chosen by the NUT. It MUST not be zero.
- An IKE_SA Responder’s SPI field set to zero.
- A Next Payload field set to SA Payload (33).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_SA_INIT (34).
- A Flags field set to (00010000)2 = (16)10.
- A Message ID field set to zero.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 4: Judgment #1

```
                                      1                   2                   3
                  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
                 ! Next     34    !0!     0       ! Length         44            !             |
                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---         |
                 !       0        !       0       ! Length         40            !  |          |
                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
                 ! Number     1   ! Prot ID    1  ! SPI Size  0   ! Trans Cnt  4 !  |          |
             --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  !       3        !       0       ! Length         8             !  |          |
    Transform |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  ! Type  1  (EN)  !       0       ! Transform ID     3    (3DES) !  |          |
             --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  !       3        !       0       ! Length         8             !  |          |SA Payload
    Transform |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |Proposal  |
              |  ! Type  2  (PR)  !       0       ! Transform ID     2    (SHA1) !  |          |
             --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  !       3        !       0       ! Length         8             !  |          |
    Transform |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  ! Type  3  (IN)  !       0       ! Transform ID     2    (SHA1) !  |          |
             --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  !       0        !       0       ! Length         8             !  |          |
    Transform |  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
              |  ! Type  4  (DH)  !       0       ! Transform ID     2    (1024) !  |          |
             --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 100 SA Payload contents

The NUT transmits an IKE_SA_INIT request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 101 SA Payload format

- A Next Payload field set to KE Payload (34).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 102 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 40 bytes for this proposal according to Common Configuration.
- A Proposal # field set to 1 if this structure is the first proposal, otherwise set to 1 greater than the previous proposal.
- A Protocol ID field set to IKE (1).
- A SPI Size field set to zero.
- A # of Transforms field set to 4.

A Transform field set to following (There are 4 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 103 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.

- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for PRF_HMAC_SHA1.
- A Transform Type field set to PRF (2).
- A RESERVED field set to zero.
- A Transform ID set to PRF_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #4

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for 1024 MODP Group.
- A Transform Type field set to D-H (4).
- A RESERVED field set to zero.
- A Transform ID set to Group2 (2).

Part C

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT request including properly formatted KE Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          DH Group #           !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Key Exchange Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 104 KE Payload format

- A Next Payload field set to Nonce Payload (40).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 136 bytes for Group 2.
- A DH Group field set to Group2 (2).
- A RESERVED field set to zero.
- A Key Exchange Data field set to Diffie-Hellman public value. The length of the Key Exchange Data field must be equal to 1024bit.

Part D

Step 8: Judgment #1

The NUT transmits an IKE_SA_INIT request including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 105 Nonce Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Nonce Data field set to random data generated by the transmitting entity. The size of the Nonce must between 16 and 256 octets.

Possible Problems:

- IKE_SA_INIT request has following packet format. It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(COOKIE)],
SA, KE, Ni,
[N(NAT_DETECTION_SOURCE_IP)+,
 N(NAT_DETECTION_DESTINATION_IP)],
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- Each of transforms can be located in the any order.

Test IKEv2.SGW.I.1.1.1.2: Sending IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #2

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link A.

Part C: IDi Payload Format (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.

Part D: AUTH Payload Format (BASIC)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link A.

Part E: SA Payload Format (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.

Part F: TSi Payload Format (BASIC)

21. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link A.
23. TN1 responds with an IKE_SA_INIT response to the NUT.
24. Observe the messages transmitted on Link A.

Part G: TSr Payload Format (BASIC)

25. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A.
27. TN1 responds with an IKE_SA_INIT response to the NUT.
28. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 106 Header format

- An IKE_SA Initiator’s SPI field set to same as the IKE_SA_INIT request’s IKE_SA

Initiator's SPI field value.

- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_AUTH (35).
- A Flags field set to (00010000)2 = (16)10.
- A Message ID field set to 1.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 107 Encrypted payload

- A Next Payload field set to IDi Payload (35).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum

must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 108 ID Payload format

- A Next Payload field set to AUTH Payload (39).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field set to ID_IPV6_ADDR (5).
- A RESERVED field set to zero.
- An Identification Data field set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 109 AUTH Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field set to Shared Key Message Integrity Code (2).
- A RESERVED field set to zero.
- An Authentication Data field set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2

```
                                    1                   2                   3
                0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
               ! Next    44    !0!      0      ! Length          40             !             |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---         |
               !       0       !       0       ! Length          36             !  |          |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
               ! Number    1   ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3   !  |          |
               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
               ! SPI value                                                      !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
             | !       3       !       0       ! Length           8             !  |          |
 Transform   | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |SA Payload
             | ! Type  1  (EN) !       0       ! Transform ID     3      (3DES) !  | Proposal |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
             | !       3       !       0       ! Length           8             !  |          |
 Transform   | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
             | ! Type  3  (IN) !       0       ! Transform ID     2      (SHA1) !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
             | !       0       !       0       ! Length           8             !  |          |
 Transform   | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
             | ! Type  5  (ESN)!       0       ! Transform ID     0        (No) !  |          |
           --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 110 SA Payload contents

The NUT transmits an IKE_AUTH request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 111 SA Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 112 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1 if this structure is the first proposal, otherwise set to 1 greater thatn the previous proposal.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 113 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 24: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 114 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 115 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- A Ending Address field set to greater thatn or equal to Prefix B.

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 116 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 117 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix Y.
- An Ending Address field set to less than or equal to Prefix Y.

Possible Problems:

- IKE_AUTH request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDi,
[CERT+],
[N(INITIAL_CONTACT)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
[IDr],
AUTH,
[CP(CFG_REQUEST)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA,
TSi,
TSr,
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- Each of transforms can be located in the any order.

- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

Test IKEv2.SGW.I.1.1.1.3: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT               TN1               TH2
(Host)           (SGW)             (SGW)             (Host)
  |                |                 |                 |
  |                |---------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                 |                 | (Judgment #1)
  |                |<----------------|                 | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |                 |                 | (Packet #1)
  |                |                 |                 |
  |                |---------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |                 |                 |                         SAi2, TSi, TSr})
  |                |                 |                 | (Judgment #2)
  |                |<----------------|                 | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                |                 |                 |                          SAr2, TSi, TSr})
  |                |                 |                 | (Packet #2)
  |                |                 |                 |
  |<---------------+=================+-----------------| IPsec {Echo Request}
  |                |                 |                 | (Packet #3) (Judgment #3)
  |----------------+=================+---------------->| IPsec {Echo Reply}
  |                |                 |                 | (Packet #4) (Judgment #4)
  |                |                 |                 |
  V                V                 V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT

6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TH2 can send Echo Reply to TH1 instead of sending Echo Request.

Group 1.2. Use of Retransmission Timers

Test IKEv2.SGW.I.1.1.2.1: Retransmissions of IKE_SA_INIT requests

Purpose:

To verify an IKEv2 device retransmits IKE_SA_INIT request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |                |
  |                * wait for the event of a timeout
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #2)
  |                |
  V                V
```

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 waits for the event of a timeout on NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request's Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.SGW.I.1.1.2.2: Stop of retransmission of IKE_SA_INIT requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Judgment #1)
  |                 |
  |                 * wait for the event of a timeout
  |                 |
  |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Judgment #2)
  |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                 | (Packet #1)
  |                 |
  |                 * wait for the event of a timeout
  |                 |
  |-------X         | never send IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Judgment #3)
  |                 |
  V                 V
```

Packet #1	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 waits for the event of a timeout on NUT.
4. Observe the messages transmitted on Link A
5. TN1 responds with an IKE_SA_INIT response to the NUT.
6. TN1 waits for the event of a timeout on NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request’s Message ID value in IKE Header.

Step 7: Judgment #3
The NUT never retransmits an IKE_SA_INIT request which has the same Message ID value as the previous IKE_SA_INIT request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.SGW.I.1.1.2.3: Retransmissions of IKE_AUTH requests

Purpose:

To verify an IKEv2 device retransmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                  TN1
 (SGW)                (SGW)
   |                    |
   |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                    | (Judgment #1)
   |<-------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                    | (Packet #1)
   |                    |
   |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                    | (Judgment #2)
   |                    |
   |                    * wait for the event of a timeout
   |                    |
   |------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                    | (Judgment #3)
   |                    |
   V                    V
```

Packet #1	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 waits for the event of a timeout on NUT.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.SGW.I.1.1.2.4: Stop of retransmission of IKE_AUTH requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. In addition, set retransmission timer to 1 second.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |                * wait for the event of a timeout
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #3)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |                * wait for the event of a timeout
  |                |
  |-------X        | never send IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #4)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 waits for the event of a timeout on NUT.

6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_AUTH response to the NUT.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Step 9: Judgment #4
The NUT never retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Group 1.3. State Synchronization and Connection Timeouts

Test IKEv2.SGW.I.1.1.3.1: State Synchronization with ICMP messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives ICMP messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
TH1             NUT            TR1            TN1            TH2
(Host)          (SGW)          (Router)       (SGW)          (Host)
 |               |              |              |              |
 |               |--------------+------------->|              | IKE_SA_INIT request (HDR, SAi1,
 |               |              |              |              |                          KEi, Ni)
 |               |              |              |              | (Judgment #1)
 |               |<-------------+--------------|              | IKE_SA_INIT Response (HDR, SAr1,
 |               |              |              |              |                          KEr, Nr)
 |               |              |              |              | (Packet #1)
 |               |              |              |              |
 |               |--------------+------------->|              | IKE_AUTH request (HDR, SK {IDi, AUTH,
 |               |              |              |              |                  SAi2, TSi, TSr})
 |               |              |              |              | (Judgment #2)
 |               |<-------------+--------------|              | IKE_AUTH Response (HDR, SK {IDr, AUTH,
 |               |              |              |              |                  SAr2, TSi, TSr})
 |               |              |              |              | (Packet #2)
 |               |              |              |              |
 |<--------------+==============+==============+--------------| IPsec {Echo Request}
 |               |              |              |              | (Packet #3) (Judgment #3)
 |---------------+==============+==============+------------->| IPsec {Echo Reply}
 |               |              |              |              | (Packet #4) (Judgment #4)
 |               |              |              |              |
 |               |<-------------|              |              | Destination Unreachable
 |               |              |              |              | (No route to destination)
 |               |              |              |              | (Packet #5)
 |               |              |              |              |
 |<--------------+==============+==============+--------------| IPsec {Echo Request}
 |               |              |              |              | (Packet #6) (Judgment #5)
 |---------------+==============+==============+------------->| IPsec {Echo Reply}
 |               |              |              |              | (Packet #7) (Judgment #6)
 |               |              |              |              |
 V               V              V              V              V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See Common Packet #21
Packet #7	See Common Packet #25

Packet #5: ICMPv6 Destination Unreachable

IPv6 Header	Source Address	TR1's Global Address on Link A
	Destination Address	NUT's Global Address on Link A
ICMPv6 Header	Type	1
	Code	0

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. After reception of an Echo Reply via NUT, TR1 transmits ICMP Destination Unreachable Message to the NUT and then TH2 transmits an Echo Request to the TH1.
11. Observe the messages transmitted on Link B.
12. TH1 transmits an Echo Reply to TH2.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT forwards an Echo Request.

Step 13: Judgment #6

The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.3.2: State Synchronization with IKE messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives IKE messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1            NUT            TN1            TH2
(Host)         (SGW)          (SGW)          (Host)
  |              |              |              |
  |              |------------->|              | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |              |              |              | (Judgment #1)
  |              |<-------------|              | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |              |              |              | (Packet #1)
  |              |              |              |
  |              |------------->|              | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |              |              |              | (Judgment #2)
  |              |<-------------|              | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |              |              |              | (Packet #2)
  |              |              |              |
  |<-------------+==============+--------------| IPsec {Echo Request}
  |              |              |              | (Packet #3) (Judgment #3)
  |--------------+==============+------------->| IPsec {Echo Reply}
  |              |              |              | (Packet #4) (Judgment #4)
  |              |              |              |
  |              |<-------------|              | INFORMATIONAL request (HDR, N(INVALID_SPI))
  |              |              |              | (Packet #5)
  |              |              |              |
  |<-------------+==============+--------------| IPsec {Echo Request}
  |              |              |              | (Packet #6) (Judgment #5)
  |--------------+==============+------------->| IPsec {Echo Reply}
  |              |              |              | (Packet #7) (Judgment #6)
  |              |              |              |
  V              V              V              V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common #25
Packet #5	See below
Packet #6	See Common Packet #21

Packet #7	See Common Packet #25

Packet #4: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link A
	Destination Address	NUT's Global Address on Link X
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. TN1 transmits INFORMATIONAL request with a Notify payload of type INVALID_ SPI to the NUT.
11. TH2 transmits an Echo Request to TH1.
12. Observe the messages transmitted on Link B.
13. TH1 transmits an Echo Reply to TH2.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 12: Judgment #5
The NUT forwards an Echo Request.

Step 14: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None

Test IKEv2.SGW.I.1.1.3.3: Close connections when repeated attempts fail

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  n each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |                * wait for the event of a timeout
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #3)
  |                |
  |                * wait for the event of a timeout
  |                |
  |-------X        | never send IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #2

Part A: (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link A.
13. TN1 waits for the event of a timeout on the NUT.
14. Observe the messages transmitted on Link A.
15. Repeat Step 5 and Step 6 until the NUT’s last restransmission comes.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Step 8: Judgment #4
The NUT never retransmits an IKE_AUTH request which has the same Message ID value as the previous IKE_AUTH request’s Message ID value in IKE Header.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.3.4: Close connections when receiving INITIAL_CONTACT

Purpose:

To verify an IKEv2 device closes connections when receiving INITIAL_CONTACT.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 7.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<---------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<---------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
  |<--------------+===============+--------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #3) (Judgment #3)
  |---------------+===============+------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #5)
  |                  |<---------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #5)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_AUTH request (HDR, SK {IDi, N(INITIAL_CONTACT),
  |                  |                  |                  |                             AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #6)
  |                  |<---------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #6)
  |                  |                  |                  |
  |          X-------+===============+--------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #7) (Judgment #7)
  |---------------+=======X            |                  | IPsec {Echo Request}
  |                  |                  |                  | (Packet #8) (Judgment #8)
  |                  |                  |                  |
  |<--------------+===============+--------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #9) (Judgment #9)
  |---------------+===============+------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #10) (Judgment #10)
  |                  |                  |                  |
  V                  V                  V                  V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet # 25
Packet #5	See Common Packet #2
Packet #6	See Common Packet #6
Packet #7	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 5.
Packet #8	See Common Packet # 25
Packet #9	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 11 to Step 14.
Packet #10	See Common Packet # 25

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. NUT transmits IKE_SA_INIT request to the NUT.
11. Observe the messages transmitted on Link A.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link A.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
15. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms.
16. Observe the messages transmitted on Link B.
17. TH1 transmits an Echo Request to TH2.
18. Observe the messages transmitted on Link A.
19. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the second negotiated algorithms.
20. Observe the messages transmitted on Link B.
21. TH1 transmits an Echo Reply to TH2.
22. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #6
The NUT transmits an IKE_AUTH request with a Notify payload of type INITIAL_CONTACT to the NUT. The IKE_AUTH request includes “ENCR_3DES”,

"AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 16: Judgment #7
The NUT never forwards an Echo Request.

Step 18: Judgment #8
The NUT never forwards an Echo Request with IPsec ESP using the first negotiated algorithms.

Step 20: Judgment #9
The NUT forwards an Echo Request.

Step 22: Judgment #10
The NUT forwards an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- **Step 18**:
  The NUT can forward an Echo Request to the TH2 with IPsec ESP using the second negotiated algorithms.

.

Test IKEv2.SGW.I.1.1.3.5: Sending Liveness check

Purpose:

To verify an IKEv2 device checks whether the other endpoint is alive.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |                * wait until receiving a liveness check
  |                |
  |--------------->| INFORMATIONAL request (HDR, SK {})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- While an INFORMATIONAL request for liveness check is transmitted, NUT needs to keep sending packets like ICMPv6 Echo Request.

Test IKEv2.SGW.I.1.1.3.6: Sending Delete Payload for IKE_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when IKE_SA is deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |                * wait until expiring IKE_SA
  |                |
  |--------------->| INFORMATIONAL request (HDR, SK {D})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.

7. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Possible Problems:

- At Step 7, NUT can transmit INFORMATIONAL request with a Delete Payload including 2 (ESP) as Protocol ID, 4 as SPI Size and SPI value to delete CHILD_SA before transmitting an INFORMATIONAL request to delete IKE_SA.

Test IKEv2.SGW.I.1.1.3.7: Sending Delete Payload for CHILD_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                 TN1
(SGW)               (SGW)
  |                   |
  |------------------>| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   | (Judgment #1)
  |<------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   | (Packet #1)
  |                   |
  |------------------>| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   | (Judgment #2)
  |<------------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   | (Packet #2)
  |                   |
  |                   * wait until expiring CHILD_SA
  |                   |
  |------------------>| INFORMATIONAL request (HDR, SK {D})
  |                   | (Judgment #3)
  |                   |
  V                   V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring CHILD_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.

7. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Possible Problems:

- None

Test IKEv2.SGW.I.1.1.3.8: Sending Liveness check with unprotected messages

Purpose:

To verify an IKEv2 device handles cryptographically unprotected Messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  Configure the timer to consider that the peer is dead to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |               | (Packet #2)
    |               |
    |<--------------| Echo Request
    |               | (Packet #3)
    |--------X      | Echo Reply
    |               | (Judgment #3)
    |               |
    |-------------->| INFORMATIONAL request (HDR, SK {})
    |               | (Judgment #4)
    |               |
    V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See below

Packet #3: Echo Request

IPv6 Header	Source Address	TN1's Global Address
	Destination Address	NUT's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	0

	Sequence Number	any
	Payload Data	0x0000000000000000

Part A: (BASIC)

8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
9. Observe the messages transmitted on Link A.
10. TN1 responds with an IKE_SA_INIT response to the NUT.
11. Observe the messages transmitted on Link A.
12. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
13. TN1 trasmits a cryptographically unprotected Echo Request to the NUT.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT never responds with a cryptographically unprotected Echo Reply. The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- NUT may have the different trigger other than timer to send an INFORMATIONAL request for the liveness check. In that case, TN must be adjusted to support such a trigger.

Group 1.4. Version Numbers and Forward Compatibility

Test IKEv2.SGW.I.1.1.4.1: Unrecognized payload types and critical bit is not set

Purpose:

To verify an IKEv2 device ignores invalid payload types when the invalid type payload's critical bit is not set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                  NUT                  TN1                  TH2
(Host)               (SGW)                (SGW)                (Host)
  |                    |                    |                    |
  |                    |------------->|                    | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    |                    |                    | (Judgment #1)
  |                    |<--------------|                    | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                    |                    |                    | (Packet #1)
  |                    |                    |                    |
  |                    |------------->|                    | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                    |                    |                    | (Judgment #2)
  |                    |<--------------|                    | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                    |                    |                    | (Packet #2)
  |                    |                    |                    |
 ...                  ...                  ...                  ...
  |                    |                    |                    |                          ---
  |<-------------+==============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                    |                    |                    | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+==============+------------->| IPsec {Echo Reply}        | is expired
  |                    |                    |                    | (Packet #4) (Judgment #4) |
  |                    |                    |                    |                          ---
 ...                  ...                  ...                  ...
  |                    |                    |                    |
  |                    |------------->|                    | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                    |                    |                    | (Judgment #5)
  |                    |                    |                    |
  |                    |<--------------|                    | CREATE_CHILD_SA response (HDR, SK {P, SA, Nr, TSi, TSr})
  |                    |                    |                    | (Packet #5)
  |                    |                    |                    |
  |<-------------+==============+--------------| IPsec {Echo Request} (new CHILD_SA)
  |                    |                    |                    | (Packet #6) (Judgment #6)
  |--------------+==============+------------->| IPsec {Echo Request} (new CHILD_SA)
  |                    |                    |                    | (Packet #7) (Judgment #7)
  |                    |                    |                    |
  V                    V                    V                    V

N: REKEY_SA
P: Payload with an invalid payload type
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 11.
Packet #7	See Common Packet #25

Packet #5: CREATE_CHILD_SA response

IPv6 Header	All fields are same as Common Packet #16 Payload	
UDP Header	All fields are same as Common Packet #16 Payload	
IKEv2 Header	All fields are same as Common Packet #16 Payload	
E payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #16	
Invalid Payload	Next Payoad	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	4

SA Payload	All fields are same as Common Packet #16 Payload
Ni, Nr paylaod	All fields are same as Common Packet #16 Payload
TSi Payload	All fields are same as Common Packet #16 Payload
TSr Payload	All fields are same as Common Packet #16 Payload

Part A: Invalid payload type 1 (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 1 and the invalid payload’s critical flag is not set.
13. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Response to the TH2.
16. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
22. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
23. Observe the messages transmitted on Link B.
24. TH1 transmits an Echo Reply to TH2.
25. Observe the messages transmitted on Link A.
26. Repeat Steps 22 through 25 until lifetime of SA is expired.
27. Observe the messages transmitted on Link A.
28. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 32 and the invalid payload’s critical flag is not set.
29. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
30. Observe the messages transmitted on Link B.
31. TH1 transmits an Echo Response to the TH2.
32. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 (BASIC)

33. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
34. Observe the messages transmitted on Link A.
35. TN1 responds with an IKE_SA_INIT response to the NUT.
36. Observe the messages transmitted on Link A.
37. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
38. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
39. Observe the messages transmitted on Link B.
40. TH1 transmits an Echo Reply to TH2.
41. Observe the messages transmitted on Link A.
42. Repeat Steps 38 through 41 until lifetime of SA is expired.
43. Observe the messages transmitted on Link A.
44. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 49 and the invalid payload's critical flag is not set.
45. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
46. Observe the messages transmitted on Link B.
47. TH1 transmits an Echo Response to the TH2.
48. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 (BASIC)

49. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
50. Observe the messages transmitted on Link A.
51. TN1 responds with an IKE_SA_INIT response to the NUT.
52. Observe the messages transmitted on Link A.
53. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
54. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
55. Observe the messages transmitted on Link B.
56. TH1 transmits an Echo Reply to TH2.
57. Observe the messages transmitted on Link A.
58. Repeat Steps 54 through 57 until lifetime of SA is expired.
59. Observe the messages transmitted on Link A.
60. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is not set.
61. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
62. Observe the messages transmitted on Link B.
63. TH1 transmits an Echo Response to the TH2.
64. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 14: Judgment #6
The NUT forwards an Echo Request to the TH1.

Step 16: Judgment #7
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part B

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 23: Judgment #3
The NUT forwards an Echo Request.

Step 25: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 27: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 30: Judgment #6
The NUT forwards an Echo Request to the TH1.

Step 32: Judgment #7
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part C

Step 34: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 36: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 39: Judgment #3
The NUT forwards an Echo Request.

Step 41: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 43: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 46: Judgment #6
The NUT forwards an Echo Request to the TH1.

Step 48: Judgment #7
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part D

Step 50: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 52: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 55: Judgment #3
The NUT forwards an Echo Request.

Step 57: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 59: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 62: Judgment #6
The NUT forwards an Echo Request to the TH1.

Step 64: Judgment #7
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.4.2: Unrecognized payload types and critical bit is set

Purpose:

To verify an IKEv2 device rejects the messages with invalid payload types when the invalid type payload's critical bit is set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                 TN1                 TH2
(Host)             (SGW)               (SGW)               (Host)
  |                  |                   |                   |
  |                  |------------->|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                   |                   | (Judgment #1)
  |                  |<--------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                   |                   | (Packet #1)
  |                  |                   |                   |
  |                  |------------->|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                   |                   | (Judgment #2)
  |                  |<--------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                   |                   | (Packet #2)
  |                  |                   |                   |
 ...                ...                 ...                 ...
  |                  |                   |                   |                            ---
  |<-------------+===============+--------------| IPsec {Echo Request}       | repeat Echo exchange
  |                  |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+------------->| IPsec {Echo Reply}         | is expired
  |                  |                   |                   | (Packet #4) (Judgment #4) |
  |                  |                   |                   |                            ---
 ...                ...                 ...                 ...
  |                  |                   |                   |
  |                  |------------->|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                   |                   | (Judgment #5)
  |                  |                   |                   |
  |                  |<--------------|                   | CREATE_CHILD_SA response (HDR, SK {P, SA, Nr, TSi, TSr})
  |                  |                   |                   | (Packet #5)
  |                  |                   |                   |
  |        X-------+===============+--------------| IPsec {Echo Request} (new CHILD_SA)
  |                  |                   |                   | (Packet #6) (Judgment #6)
  |--------------+=======X           |                   | IPsec {Echo Request} (new CHILD_SA)
  |                  |                   |                   | (Packet #7) (Judgment #7)
  |                  |                   |                   |
  V                  V                   V                   V

N: REKEY_SA
P: Payload with an invalid payload type
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 11.
Packet #7	See Common Packet #25

Packet #5: CREATE_CHILD_SA response

IPv6 Header		All fields are same as Common Packet #16 Payload
UDP Header		All fields are same as Common Packet #16 Payload
IKEv2 Header		All fields are same as Common Packet #16 Payload
E payload	Next Payload	*Invalid payload type value*
		Other fields are same as Common Packet #16
Invalid Payload	Next Payoad	33 (SA)
	Critical	1
	Reserved	0
	Payload Length	4
SA Payload		All fields are same as Common Packet #16 Payload
Ni, Nr paylaod		All fields are same as Common Packet #16 Payload

TSi Payload	All fields are same as Common Packet #16 Payload
TSr Payload	All fields are same as Common Packet #16 Payload

Part A: Invalid payload type 1 and Critical bit is set (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 1 and the invalid payload’s critical flag is set.
13. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Response to the TH2.
16. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 and Critical bit is set (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
22. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
23. Observe the messages transmitted on Link B.
24. TH1 transmits an Echo Reply to TH2.
25. Observe the messages transmitted on Link A.
26. Repeat Steps 22 through 25 until lifetime of SA is expired.
27. Observe the messages transmitted on Link A.
28. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 32 and the invalid payload’s critical flag is set.
29. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
30. Observe the messages transmitted on Link B.
31. TH1 transmits an Echo Response to the TH2.
32. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 Critical bit is set (BASIC)

33. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
34. Observe the messages transmitted on Link A.
35. TN1 responds with an IKE_SA_INIT response to the NUT.
36. Observe the messages transmitted on Link A.
37. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
38. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
39. Observe the messages transmitted on Link B.
40. TH1 transmits an Echo Reply to TH2.
41. Observe the messages transmitted on Link A.
42. Repeat Steps 38 through 41 until lifetime of SA is expired.
43. Observe the messages transmitted on Link A.
44. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 49 and the invalid payload's critical flag is set.
45. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
46. Observe the messages transmitted on Link B.
47. TH1 transmits an Echo Response to the TH2.
48. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 Critical bit is set (BASIC)

49. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
50. Observe the messages transmitted on Link A.
51. TN1 responds with an IKE_SA_INIT response to the NUT.
52. Observe the messages transmitted on Link A.
53. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
54. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
55. Observe the messages transmitted on Link B.
56. TH1 transmits an Echo Reply to TH2.
57. Observe the messages transmitted on Link A.
58. Repeat Steps 54 through 57 until lifetime of SA is expired.
59. Observe the messages transmitted on Link A.
60. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which includes a payload with invalid payload type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is set.
61. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
62. Observe the messages transmitted on Link B.
63. TH1 transmits an Echo Response to the TH2.
64. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 14: Judgment #6
The NUT never forwards an Echo Request to the TH1.

Step 16: Judgment #7
The NUT never forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part B

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 23: Judgment #3
The NUT forwards an Echo Request.

Step 25: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 27: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 30: Judgment #6
The NUT never forwards an Echo Request to the TH1.

Step 32: Judgment #7

The NUT never forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part C

Step 34: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 36: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 39: Judgment #3
The NUT forwards an Echo Request.

Step 41: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 43: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 46: Judgment #6
The NUT never forwards an Echo Request to the TH1.

Step 48: Judgment #7
The NUT never forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Part D

Step 50: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 52: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 55: Judgment #3
The NUT forwards an Echo Request.

Step 57: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 59: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 62: Judgment #6
The NUT never forwards an Echo Request to the TH1.

Step 64: Judgment #7
The NUT never forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Possible Problems:

- None.

Group 1.5. Cookies

Test IKEv2.SGW.I.1.1.5.1: Retrying IKE_SA_INIT request with a Notify payload of type COOKIE

Purpose:

To verify an IKEv2 device retries IKE_SA_INIT request using a Notify payload of type COOKIE.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(SGW)                (SGW)
  |                    |
  |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    | (Judgment #1)
  |<-------------------| IKE_SA_INIT Response (HDR, N(COOKIE))
  |                    | (Packet #1)
  |                    |
  |------------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
  |                    | (Judgment #2)
  |                    |
  V                    V
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header		All fields are same as Common Packet #2
UDP Header		All fields are same as Common Packet #2
IKEv2 Header	IKE_SA Initiator's SPI	The same value as corresponding request's IKE_SA Initiator's SPI value
	IKE_SA Responder's SPI	0
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	34 (IKE_SA_INIT)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0

	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.
4. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_SA_INIT request including a Notify payload of type COOKIE containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 118 Notify Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A SPI Size field set to zero.
- A Notify Message Type field set to COOKIE (16390).
- A Notification Data field set to the TN1 supplied cookie data.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.5.2: Interaction of COOKIE and INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles a series of the Initial Exchanges using a Notify payload of type COOKIE and type INVALID_KE_PAYLOAD.

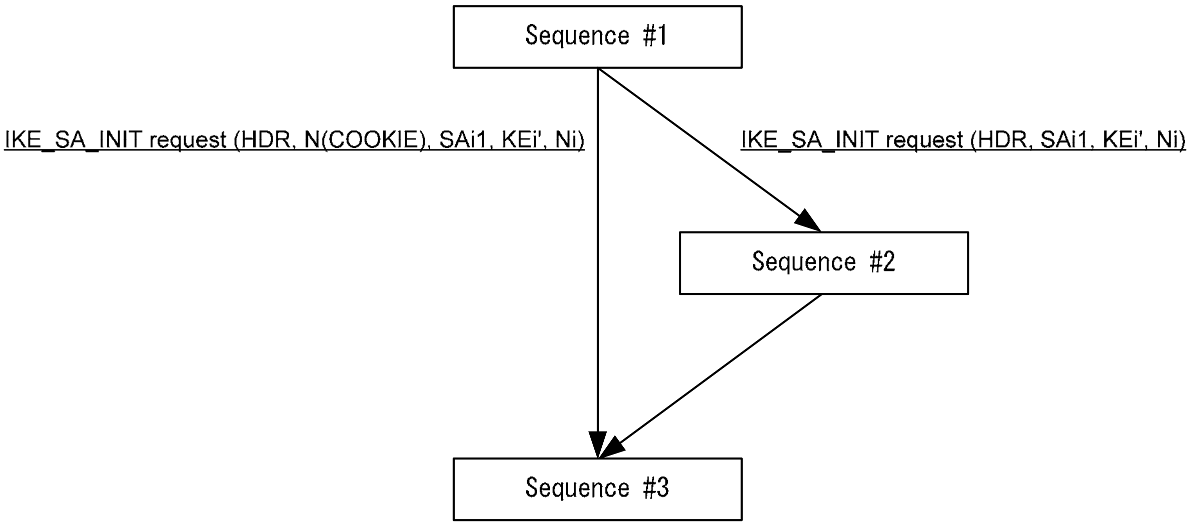
References:

- [RFC 4306] - Sections 2.6, 2.7 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:



```
Sequence #1:
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Packet #1)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Judgment #2)
    |<---------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Packet #2)
    |                |
    |------*1------->| IKE_SA_INIT request (HDR, SAi1, KEi', Ni)
    |                |   or
    |------*2------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi', Ni)
    |                | (Judgment #3)
    |                |
    *1) If the NUT send IKE_SA_INIT request (HDR, SAi1, KEi', Ni), go to Sequence #2.
    *2) If the NUT send IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi', Ni), go to Sequence #3.
    Otherwise, this test is failed.

Sequence #2:
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |<---------------| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                | (Packet #3)
    |                |
    |--------------->| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi', Ni)
    |                | (Judgment #4)
    |                |
    Go to Sequence #3.

Sequence #3:
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #4)
    |                |
    |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #2

Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #2: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #3: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Different cookie value from Packet #1's cookie value.
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD to the NUT.

6. Observe the messages transmitted on Link A.
7. If the IKE_SA_INIT request from NUT includes a Notify payload of type COOKIE, TN1 responds with an IKE_SA_INIT response. The message has a different cookie value from the cookie value at Step3.
   A) Observe the messages transmitted on Link A.
   B) TN1 responds with an IKE_SA_INIT response.
8. If the IKE_SA_INIT request from NUT does not include a Notify payload of type COOKIE, TN1 responds with an IKE_SA_INIT response
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request. The message has a Notify payload of type COOKIE with the cookie data supplied by the responder as the first payload. All other payloads are unchanged.

Step 6: Judgment #3
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 5. All other payloads are unchanged.

Step 7A: Judgment #4
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 7. All other payloads are unchanged.

Step 9: Judgment #5
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Test IKEv2.SGW.I.1.1.5.3: Interaction of COOKIE and INVALID_KE_PAYLOAD with unoptimized Responder

Purpose:

To verify an IKEv2 device properly handles a series of the Initial Exchanges using a Notify ayload of type COOKIE and type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.6, 2.7 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Judgment #1)
    |<--------------| IKE_SA_INIT response (HDR, N(COOKIE))
    |                | (Packet #1)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |                | (Judgment #2)
    |<--------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |                | (Packet #2)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi ' , Ni)
    |                |   or
    |-------------->| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni)
    |                | (Judgment #3)
    |                |
    |<--------------| IKE_SA_INIT response (HDR, N(COOKIE'))
    |                | (Packet #3)
    |                |
    |-------------->| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi ' , Ni)
    |                | (Judgment #4)
    |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Packet #4)
    |                |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                | (Judgment #5)
    |                |
    V                V
```

Packet #1	See below
Packet #2	See below

Packet #3	See below
Packet #4	See Common Packet #2

Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Cookie value
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #2: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Packet #3: IKE_SA_INIT request

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	Different cookie value from Packet #1's cookie value.
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response including a Notify payload of type COOKIE to the NUT.

4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD to the NUT.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response. The message has a different cookie value from the cookie value at Step3.
8. Observe the messages transmitted on Link A.
9. TN1 responds with an IKE_SA_INIT response.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request. The message has a Notify payload of type COOKIE with the cookie data supplied by the responder as the first payload. All other payloads are unchanged.

Step 6: Judgment #3
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 5.

Step 8: Judgment #4
The NUT transmits an IKE_SA_INIT request including a Key Exchange payload which contains a recalculated Key Exchange Data. The message can have a Notify payload of type COOKIE with the cookie data supplied by the responder at Step 7. All other payloads are unchanged.

Step 10: Judgment #5
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- None.

Group 1.6. Cryptographic Algorithm Negotiation

Test IKEv2.SGW.I.1.1.6.1: Cryptographic Algorithm Negotiation for IKE_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part E, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	***ENCR_AES_CBC***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	***ENCR_AES_CTR***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	***PRF_AES128_CBC***	AUTH_HMAC_SHA1_96	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	***AUTH_AES_XCBC_96***	Group 2
Part E	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  | (Packet #1)
  |                  |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See Common Packet #2

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.

4. Observe the messages transmitted on Link B.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link B.
7. TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link B.

Part C: Pseudo-Random Function PRF_AES128_CBC (ADVANCED)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link B.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link B.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link B.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link B.

Part E: D-H Group Group 14 (ADVANCED)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link B.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_AES_CBC", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 1.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_AES_CTR", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 5.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_AES128_CBC", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 13.

Part D

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_AES_XCBC_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 17.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as proposed algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH request which is cryptographically protected by the proposed algorithms in Step 25.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.2: Cryptographic Algorithm Negotiation for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part F, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**Extended Sequence Numbers**
Part A	***ENCR_AES_CBC***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part B	***ENCR_AES_CTR***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part C	***ENCR_NULL***	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part D	ENCR_3DES	***AUTH_AES_XCBC_96***	No Extended Sequence Numbers
Part E	ENCR_3DES	***NONE***	No Extended Sequence Numbers
Part F	ENCR_3DES	AUTH_HMAC_SHA1_96	***Extended Sequence Numbers***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
TH1          NUT          TN1          TH2
(Host)       (SGW)        (SGW)        (Host)
  |            |            |            |
  |            |----------->|            | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |            |            |            | (Judgment #1)
  |            |<-----------|            | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |            |            |            | (Packet #1)
  |            |            |            |
  |            |----------->|            | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |            |            |            |                          SAi2, TSi, TSr})
  |            |            |            | (Judgment #2)
  |            |<-----------|            | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |            |            |            |                          SAr2, TSi, TSr})
  |            |            |            | (Packet #2)
  |            |            |            |
  |<-----------+============+------------| IPsec {Echo Request}
  |            |            |            | (Packet #3) (Judgment #3)
  |------------+============+----------->| IPsec {Echo Reply}
  |            |            |            | (Packet #3) (Judgment #4)
  |            |            |            |
  V            V            V            V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

10. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
11. Observe the messages transmitted on Link B.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link B.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
15. TH2 transmits an Echo Request to TH1.
16. Observe the messages transmitted on Link A.
17. TH1 transmits an Echo Reply to TH2.
18. Observe the messages transmitted on Link B.

Part C: Encryption Algorithm ENCR_NULL (ADVANCED)

19. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link B.
21. TN1 responds with an IKE_SA_INIT response to the NUT.
22. Observe the messages transmitted on Link B.
23. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
24. TH2 transmits an Echo Request to TH1.
25. Observe the messages transmitted on Link A.
26. TH1 transmits an Echo Reply to TH2.
27. Observe the messages transmitted on Link B.

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

28. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
29. Observe the messages transmitted on Link B.
30. TN1 responds with an IKE_SA_INIT response to the NUT.
31. Observe the messages transmitted on Link B.
32. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
33. TH2 transmits an Echo Request to TH1.
34. Observe the messages transmitted on Link A.
35. TH1 transmits an Echo Reply to TH2.
36. Observe the messages transmitted on Link B.

Part E: Integrity Algorithm NONE (ADVANCED)

37. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

38. Observe the messages transmitted on Link B.
39. TN1 responds with an IKE_SA_INIT response to the NUT.
40. Observe the messages transmitted on Link B.
41. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
42. TH2 transmits an Echo Request to TH1.
43. Observe the messages transmitted on Link A.
44. TH1 transmits an Echo Reply to TH2.
45. Observe the messages transmitted on Link B.

Part F: Extended Sequence Numbers (ADVANCED)

46. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
47. Observe the messages transmitted on Link B.
48. TN1 responds with an IKE_SA_INIT response to the NUT.
49. Observe the messages transmitted on Link B.
50. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
51. TH2 transmits an Echo Request to TH1.
52. Observe the messages transmitted on Link A.
53. TH1 transmits an Echo Reply to TH2.
54. Observe the messages transmitted on Link B.
55. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 11: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_AES_CTR”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 16: Judgment #3
The NUT forwards an Echo Request.

Step 18: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_NULL”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 25: Judgment #3
The NUT forwards an Echo Request.

Step 27: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part D

Step 29: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 31: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 34: Judgment #3
The NUT forwards an Echo Request.

Step 36: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part E

Step 38: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 40: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “NONE” and “No Extended Sequence Numbers” as proposed algorithms.

Step 43: Judgment #3
The NUT forwards an Echo Request.

Step 45: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part F

Step 47: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 49: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1” and “Extended Sequence Numbers” as proposed algorithms.

Step 52: Judgment #3
The NUT forwards an Echo Request.

Step 54: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.3: Sending Multiple Transforms for IKE_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_SA_INIT request with multiple transforms for IKE_SA.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_HMAC_SHA1** **PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 2** **Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(SGW)                (SGW)
  |                    |
  |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    | (Judgment #1)
  V                    V
```

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link B.

Part B: Multiple Pseudo-Random Functions (ADVANCED)

3. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link B.

Part C: Multiple Integrity Algorithms (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above.

6. Observe the messages transmitted on Link B.

Part D: Multiple D-H Groups (ADVANCED)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
8. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “PRF_AES128_CBC“AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “D-H group 2” and “D-H group 14” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.4: Sending Multiple Proposals for IKE_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple proposals for CHILD_SA.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_SA_INIT exchanges Algorithms					
	Proposal	**Protocol ID**	**Encryption**	**PRF**	**Integrity**	**D-H Group**
Part A	Proposal #1	IKE	ENCR_ 3DES	PRF_ HMAC_SHA1	AUTH_ HMAC_SHA1_96	Group 2
	Proposal #2	**IKE**	**ENCR_ AES_CBC**	**PRF_ AES128_CBC**	**AUTH_ AES_XCBC_96**	**Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(SGW)                (SGW)
  |                    |
  |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    | (Judgment #1)
  |                    |
  V                    V
```

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” in SA Proposal #1 (ESP) and “ENCR_AES_CBC”, “PRF_AES128_CBC”, “AUTH_AES_XCBC_96” and “D-H group 14” in SA Proposal #2 (ESP) as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.5: Sending Multiple Transforms for CHILD_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple transforms for CHILD_SA.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |----------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Judgment #1)
  |<-----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  | (Packet #1)
  |                  |
  |----------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See Common Packet #2

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
2. Observe the messages transmitted on Link B.
3. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.
4. Observe the messages transmitted on Link B.

Part B: Multiple Integrity Algorithms (ADVANCED)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
6. Observe the messages transmitted on Link B.
7. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.
8. Observe the messages transmitted on Link B.

Part C: Extended Sequecnce Numbers (ADVANCED)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request including a SA payload as described above to the TN1.
10. Observe the messages transmitted on Link B.
11. NUT transmits an IKE_AUTH request including a SA payload as described above to the TN1.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “No Extended Sequence Numbers” and “Extended Sequence Number” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.6: Sending Multiple Proposals for CHILD_SA

Purpose:

To verify an IKEv2 device properly transmits IKE_AUTH request with multiple proposals for CHILD_SA.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IKE_AUTH exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	**ESP**	**ENCR_AES_CBC**	**AUTH_AES_XCBC_96**	**ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Judgment #1)
  |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                 | (Packet #1)
  |                 |
  |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  V                 V
```

Packet #1	See Common Packet #2

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" in SA Proposal #1 (ESP) and then "ENCR_AES_CBC", "AUTH_AES_XCBC_96" and "Extended Sequence Numbers" in SA Proposal #2 (ESP) as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.7: Receipt of INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  TH1                NUT                TN1                TH2
 (Host)             (SGW)              (SGW)              (Host)
   |                  |                  |                  |
   |                  |----------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                  |                  |                  | (Judgment #1)
   |                  |<-----------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                  |                  |                  | (Packet #1)
   |                  |                  |                  |
   |                  |----------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                  |                  |                  | (Judgment #2)
   |                  |<-----------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                  |                  |                  | (Packet #2)
   |                  |                  |                  |
...                ...                ...                ...
   |                  |                  |                  |                              ---
   |<-----------------+==================+------------------| IPsec {Echo Request}        | repeat Echo Exchange
   |                  |                  |                  | (Packet #3) (Judgment #3)   | until lifetime of
   |------------------+==================+----------------->| IPsec {Echo Reply}          | SA is expired
   |                  |                  |                  | (Packet #4) (Judgment #4)   |
   |                  |                  |                  |                              ---
 ...                ...                ...                ...
   |                  |                  |                  |
   |                  |----------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi,
TSr})
   |                  |                  |                  | (Judgment #4)
   |                  |<-----------------|                  | CREATE_CHILD_SA response (HDR, SK,
N(INVALID_KE_PAYLOAD)})
   |                  |                  |                  | (Packet #5)
   |                  |                  |                  |
   |                  |----------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi', TSi,
TSr})
   |                  |                  |                  | (Judgment #5)
   |                  |                  |                  |
   V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below

Packet #5: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #16
UDP Header		Same as Common Packet #16
IKEv2 Header		Same as Common Packet #16
E Payload		Same as Common Packet #16
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.

3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link B.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT.
11. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.8: Receipt of NO_PROPOSAL_CHOSEN

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT response with a Notify payload of type NO_PROPOSAL_CHOSEN.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |----------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<-----------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |----------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<-----------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
...                ...                ...                ...
  |                  |                  |                  |                              ---
  |<-----------------+==================+------------------| IPsec {Echo Request}        | repeat Echo Exchange
  |                  |                  |                  | (Packet #3) (Judgment #3)   | until lifetime of
  |------------------+==================+----------------->| IPsec {Echo Reply}          | SA is expired
  |                  |                  |                  | (Packet #4) (Judgment #4)   |
  |                  |                  |                  |                              ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |----------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi,
TSr})
  |                  |                  |                  | (Judgment #4)
  |                  |<-----------------|                  | CREATE_CHILD_SA response (HDR, SK,
N(INVALID_KE_PAYLOAD)})
  |                  |                  |                  | (Packet #5)
  |                  |                  |                  |
  |                  |----------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi', TSi,
TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below

Packet #5: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #16
UDP Header		Same as Common Packet #16
IKEv2 Header		Same as Common Packet #16
E Payload		Same as Common Packet #16
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.

3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link B.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type NO_PROPOSAL_CHOSEN to the NUT.
11. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. The new CREATE_CHILD_SA request is not a retransmitted request.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.9: Response with inconsistent SA proposal for IKE_SA

Purpose:

To verify an IKEv2 device properly handles a response with a SA payload which is inconsistent with one of its proposals.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT                 TN1
  (SGW)               (SGW)
    |                   |
    |------------------>| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                   | (Judgment #1)
    |<------------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
    |                   | (Packet #1)
    |                   |
    |-------X           | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                   | (Judgment #2)
    |                   |
    V                   V
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header	Same as the Common Packet #2
UDP Header	Same as the Common Packet #2
IKEv2 Header	Same as the Common Packet #2
SA Payload	See below
KEi Payload	Same as the Common Packet #2
Ni Payload	Same as the Common Packet #2

SA Payload	Next Payload			34 (KE)
	Critical			0
	Reserved			0
	Payload Length			44
	Proposal #1	SA Proposal	Next Payload	0 (last)
			Reserved	0
			Proposal Length	40
			Proposal #	1
			Protocol ID	1 (IKE)
			SPI Size	0
			# of Transforms	4

			SA Transform		See below
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	2 (PRF)
				Reserved	0
				Transform ID	2 (HMAC_SHA1)
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	4 (D-H)
				Reserved	0
				Transform ID	2 (1024 MODP Group)

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		12
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT. But the response includes a SA payload which has a different Transform ID from the proposed one.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT never transmits an IKE_AUTH request.

Possible Problems:

- Step 4
  The NUT may transmit or retransmit an IKE_SA_INIT request.

Test IKEv2.SGW.I.1.1.6.10: Response with inconsistent proposal for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a response with a SA payload which is inconsistent with one of its proposals.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |--------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #1)
  |                |<---------------|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #1)
  |                |                |                |
  |                |--------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |                |                |                          SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |<---------------|                | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                |                |                |                           SAr2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |                |                |
  |        X-------+================+----------------| IPsec {Echo Request}
  |                |                |                | (Packet #3) (Judgment #3)
  |----------------+----------------+--------------->| IPsec {Echo Reply}
  |                |                |                | (Packet #4) (Judgment #4)
  |                |                |                |
  V                V                V                V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #19

Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #6
UDP Header	Same as the Common Packet #6
IKEv2 Header	Same as the Common Packet #6
E Payload	Same as the Common Packet #6
IDr Payload	Same as the Common Packet #6

AUTH Payload	Same as the Common Packet #6
N Payload	Same as the Common Packet #6
SA Payload	See below
TSi Payload	Same as the Common Packet #6
TSr Payload	Same as the Common Packet #6

SA Payload	Next Payload				44 (TSi)
	Critical				0
	Reserved				0
	Payload Length				44
	Proposal #1	SA Proposal	Next Payload		0 (last)
			Reserved		0
			Proposal Length		40
			Proposal #		1
			Protocol ID		3 (ESP)
			SPI Size		4
			# of Transforms		3
			SA Transform		See below
			SA Transform	Next Payload	3 (more)
				Reserved	0
				Transform Length	8
				Transform Type	3 (INTEG)
				Reserved	0
				Transform ID	2 (HMAC_SHA1_96)
			SA Transform	Next Payload	0 (last)
				Reserved	0
				Transform Length	8
				Transform Type	5 (Extended Sequence Number)
				Reserved	0
				Transform ID	0 (No Extended Sequence Number)

SA Transform	Next Payload		3 (more)
	Reserved		0
	Transform Length		12
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 responds with an IKE_AUTH response to the NUT. But the response includes a SA payload which has a different Transform ID from the proposed one.
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT never forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an unprotected Echo Reply.

Possible Problems:

- Step 7
  The NUT may transmit or retransmit an IKE_AUTH request. And the NUT may notify INVALID_SPI.

Test IKEv2.SGW.I.1.1.6.11: Receipt of INVALID_KE_PAYLOAD in Initial Exchange

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT Response with a Notify payload of type INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Judgment #1)
   |<---------------| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
   |                | (Packet #1)
   |                |
   |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi', Ni)
   |                | (Judgment #2)
   |                |
   V                V
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header		Same as Common Packet #2
UDP Header		Same as Common Packet #2
IKEv2 Header		Same as Common Packet #2
	IKE_SA Responder's SPI	See each Part
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	INVALID_KE_PAYLOAD (17)
	Notification Data	The accepted D-H Group # (2)

Part A: IKE_SA Responder's SPI is zero (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

10. Observe the messages transmitted on Link A.
11. TN1 responds with an IKE_SA_INIT Response including a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT. The message's IKE_SA Responder's SPI is set to zero.
12. Observe the messages transmitted on Link A.

Part B: IKE_SA Responder's SPI is not zero (BASIC)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. TN1 responds with an IKE_SA_INIT Response including a Notify payload of type INVALID_KE_PAYLOAD containing 2 (1024 Bit MODP) as Notification Data to the NUT. The message's IKE_SA Responder's SPI is set to one.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Part B

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT Request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.6.12: Creating an IKE_SA without a CHILD_SA

Purpose:

To verify an IKEv2 device can handles a failure of creating a CHILD_SA during the IKE_AUTH exchange.

References:

- [RFC 4718] - Sections 4.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                TN1
 (SGW)              (SGW)
   |                  |
   |----------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                  | (Judgment #1)
   |<-----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                  | (Packet #1)
   |                  |
   |----------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                  | (Judgment #2)
   |<-----------------| IKE_AUTH Response (HDR, N(NO_PROPOSAL_CHOSEN))
   |                  | (Packet #2)
   |                  |
   |<-----------------| INFORMATIONAL request (HDR, SK {})
   |                  | (Packet #3)
   |----------------->| INFORMATIONAL response (HDR, SK {})
   |                  | (Judgment #3)
   |                  |
   V                  V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #17

Packet #4: IKE_AUTH response

IPv6 Header		Same as Common Packet #6
UDP Header		Same as Common Packet #6
IKEv2 Header		Same as Common Packet #6
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0

	Notify Message Type	NO_PROPOSAL_CHOSEN (14)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response with a Notify payload of type NO_PROPOSAL_CHOSEN to the NUT.
6. TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Group 1.7. Traffic Selector Negotiation

Test IKEv2.SGW.I.1.1.7.1: Narrowing the range of members of the set of traffic selectors

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2              TH3
(Host)           (SGW)            (SGW)            (Host)           (Host)
  |                |                |                |                |
  |                |-------------->|                |                | IKE_SA_INIT request (HDR, SAi1,
  |                |                |                |                |                         KEi, Ni)
  |                |                |                |                | (Judgment #1)
  |                |<--------------|                |                | IKE_SA_INIT Response (HDR, SAr1,
  |                |                |                |                |                          KEr, Nr)
  |                |                |                |                | (Packet #1)
  |                |                |                |                |
  |                |-------------->|                |                | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |                |                |                |                  SAi2, TSi, TSr})
  |                |                |                |                | (Judgment #2)
  |                |<--------------|                |                | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |                |                |                |                |                   SAr2, TSi, TSr})
  |                |                |                |                | (Packet #2)
  |                |                |                |                |
  |<--------------+==============+--------------|                | IPsec {Echo Request}
  |                |                |                |                | (Packet #3) (Judgment #3)
  |--------------+==============+-------------->|                | IPsec {Echo Reply}
  |                |                |                |                | (Packet #4) (Judgment #4)
  |                |                |                |                |
  |        X-------+==============+-----------------------------| IPsec {Echo Request}
  |                |                |                |                | (Packet #5) (Judgment #5)
  |------------------------------------------------------------->| IPsec {Echo Request}
  |                |                |                |                | (Packet #6) (Judgment #6)
  |                |                |                |                |
  V                V                V                V                V
```

Packet #1	See Common Packet #2

Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below

Packet #5: ICMPv6 Echo Request

IPv6 Header	Same as Common Packet #21	
ESP	Same as Common Packet #21	
IPv6 Header	Source Address	TH3's Global Address
	Other fields are same as Common Packet #21	
ICMPv6 Header	Same as Common Packet #21	

Packet #6: ICMPv6 Echo Reply

IPv6 Header	Source Address	TH1's Global Address
	Destination Address	TH3's Global Address
ICMPv6 Header	Same as Common Packet #25	

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TH2 transmits an Echo Request packet to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply packet to TH2.
9. Observe the messages transmitted on Link B.
10. TH3 transmits an Echo Request to TH1.
11. Observe the messages transmitted on Link A.
12. TH1 transmits an Echo Request to TH3.
13. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Request with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5

The NUT never forwards an Echo Request.

Step 13: Judgment #6
The NUT forwards an Echo Request without IPsec ESP.

Possible Problems:

- None.

Group 1.8. Error Handling

Test IKEv2.SGW.I.1.1.8.1: INVALID_IKE_SPI

Purpose:

To verify an IKEv2 device properly handles an unrecognized destination SPI.

References:

- [RFC 4306] - Sections 2.21 and 3.10.1
- [RFC 4718] - Sections 7.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |-------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #1)
  |                |<--------------|                | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #1)
  |                |                |                |
  |                |-------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |<--------------|                | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |                |                |
...              ...              ...              ...
  |                |                |                |                              ---
  |<--------------+===============+---------------| IPsec {Echo Request}      | repeat Echo Exchange
  |                |                |                | (Packet #3) (Judgment #3) | until lifetime of
  |---------------+===============+-------------->| IPsec {Echo Reply}        | SA is expired
  |                |                |                | (Packet #4) (Judgment #4) |
  |                |                |                |                              ---
 ...              ...              ...              ...
  |                |                |                |
  |                |-------------->|                | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi,
TSr})
  |                |                |                | (Judgment #4)
  |                |<--------------|                | CREATE_CHILD_SA response (HDR, SK {SA, Nr, KEr, TSi,
TSr})
  |                |                |                | (Packet #5)
  |                |                |                |
  |                |-------------->|                | INFORMATIONAL request (HDR, N(INVALID_IKE_SPI))
  |                |                |                | (Judgment #5)
  |                |                |                |
  V                V                V                V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below

Part A

Packet #5: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #16
UDP Header		Same as Common Packet #16
IKEv2 Header	IKE_SA Initator's SPI	The IKE_SA Initiator's SPI value used by this IKE message plus 1
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
	Other field are same as Common Packet #16	
E Payload		Same as Common Packet #16
N Payload		Same as Common Packet #16
SA Payload		Same as Common Packet #16
Ni, Nr Payload		Same as Common Packet #16
TSi Payload		Same as Common Packet #16
TSr Payload		Same as Common Packet #16

Part B

Packet #5: CREATE_CHILD_SA response

IPv6 Header		Same as Common Packet #16
UDP Header		Same as Common Packet #16
IKEv2 Header	IKE_SA Initator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message plus 1
		Other field are same as Common Packet #16
E Payload		Same as Common Packet #16
N Payload		Same as Common Packet #16
SA Payload		Same as Common Packet #16
Ni, Nr Payload		Same as Common Packet #16
TSi Payload		Same as Common Packet #16
TSr Payload		Same as Common Packet #16

Part A: Different IKE Initiator's SPI (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. Repeat Steps 6 and 7 until lifetime of SA is expired.
9. Observe the messages transmitted on Link B.
10. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which has an invalid value as IKE_SA Initiator's SPI to the NUT.
11. Observe the messages transmitted on Link B.

Part B: Different IKE Responder's SPI (BASIC)

12. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
13. Observe the messages transmitted on Link B.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link B.
16. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
17. TN1 transmits an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
18. Observe the messages transmitted on Link B.
19. Repeat Steps 6 and 7 until lifetime of SA is expired.
20. Observe the messages transmitted on Link B.
21. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response which has an invalid value as IKE_SA Responder's SPI to the NUT.
22. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 9: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 11: Judgment #5
NUT does not any packets or may transmit an INFORMATIONAL request with a Notify payload of type INVALID_IKE_SPI.

Part B

Step 13: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 15: Judgment #2
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms and a Key Exchange payload which contains a recalculated Key Exchange Data.

Step 18: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using corresponding algorithms.

Step 20: Judgment #4
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 22: Judgment #5
NUT does not any packets or may transmit an INFORMATIONAL request with a Notify payload of type INVALID_IKE_SPI.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.8.2: INVALID_SELECTORS

Purpose:

To verify an IKEv2 device properly handles an ESP or AH packet whose selectors do not match those of the CHILD_SA.

References:

- [RFC 4306] - Sections 2.21
- [RFC 4307] - Sections 7.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	TSi	TSr
IP Protocol ID	***IPv6-ICMP***	***IPv6-ICMP***
Start Port	0	0
End Port	65535	65535
Starting Address	TH1	NUT
Ending Address	TH1	NUT

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |----------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<-----------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |----------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, ,
  |                  |                  |                  |                       SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<-----------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |                  |                  |                  |                       SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
  |         X--------+==================+------------------| IPsec {TCP SYN}
  |                  |                  |                  | (Packet #3)
  |                  |-------X          |                  | No packets
  |                  |      or          |                  |
  |                  |----------------->|                  | INFORMATIONAL request (HDR, SK {N(INVALID_SELECTORS)})
  |                  |                  |                  | (Judgment #3)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #5) (Judgment #5)
  |                  |                  |                  |
  V                  V                  V                  V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #21
Packet #5	See Common Packet #25

Packet #2: IKE_AUTH response

TSi Payload			
	Traffic Selector	IP Protocol ID	58 (IPV6-ICMP)
		Other fields are same as Common Packet #6	

TSr Payload			
	Traffic Selector	IP Protocol ID	58 (IPV6-ICMP)
		Other fields are same as Common Packet #6	

Packet #3: TCP-SYN

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	6 (TCP)
	Integrity Check Value	The cryptographic checksum of the entire message
TCP Header	Source Port	30000
	Destination Port	30000
	Flags	SYN (0x02)

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits a TCP-SYN packet to TH1.
7. TN1 encapsulates a TCP-SYN packet with IPsec ESP using algorithms negotiated at between Step 1 and Step 5, though an Echo Request does not match the selector on TN1.
8. Observe the messages transmitted on Link B.
9. TH2 transmits an Echo Reply to TH1.
10. Observe the messages transmitted on Link A.
11. TH1 transmits an Echo Reply to TH2.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 8: Judgment #3
The NUT does not transmit any packets or transmits an INFORMATIONAL request with a Notify of type INVALID_SELECTORS.

Step 10: Judgment #4
The NUT forwards an Echo Reply.

Step 12: Judgment #5
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- Notification Type depends on the implementation at Step 8.
- If the NUT uses TCP port 30000 for other applications, the TN1 transmits TCP-SYN packets to other closed TCP port on the NUT.

Group 1.9 ID payload

Test IKEv2.SGW.I.1.1.9.1: Sending IDi Payload

Purpose:

To verify an IKEv2 device transmits IDi payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDi payload which NUT sends to TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #2

Part A: ID_IPV4_ADDR (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.

Part B: ID_FQDN (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link B.
7. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
8. Observe the messages transmitted on Link B.

Part C: ID_RFC822_ADDR (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link B.
11. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.9.2: Receiving IDr Payload

Purpose:

To verify an IKEv2 device receives IDr payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDr payload which NUT receives from TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<------------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<------------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
  |<------------------+===================+-------------------| IPsec {Echo Reply}
  |                   |                   |                   | (Packet #3) (Judgment #3)
  |-------------------+===================+------------------>| IPsec {Echo Reply}
  |                   |                   |                   | (Packet #4) (Judgment #4)
  |                   |                   |                   |
  V                   V                   V                   V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Packet #2: IKE_AUTH response

IPv6 Header		Same as Common Packet #6
UDP Header		Same as Common Packet #6
IKEv2 Header		Same as Common Packet #6
E Payload		Same as Common Packet #6
IDi Payload	ID Type	Depends on each test
	Identification Data	Depends on each test
		Same as Common Packet #6
AUTH Payload		Same as Common Packet #6
N Payload		Same as Common Packet #6
SA Payload		Same as Common Packet #6
TSi Payload		Same as Common Packet #6
TSr Payload		Same as Common Packet #6

Part A: ID_IPV4_ADDR (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDi payload as describe above to the NUT
6. TH2 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH2.
9. Observe the messages transmitted on Link B.

Part B: ID_FQDN (BASIC)

10. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
11. Observe the messages transmitted on Link B.
12. TN1 responds with an IKE_SA_INIT response to the NUT.
13. Observe the messages transmitted on Link B.
14. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDi payload as describe above to the NUT
15. TH2 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH1.
16. Observe the messages transmitted on Link B.
17. TH1 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH2.
18. Observe the messages transmitted on Link B.

Part C: ID_RFC822_ADDR (BASIC)

19. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link B.
21. TN1 responds with an IKE_SA_INIT response to the NUT.
22. Observe the messages transmitted on Link B.
23. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDi payload as describe above to the NUT
24. TH2 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH1.
25. Observe the messages transmitted on Link B.
26. TH1 transmits an Echo Reply with IPsec ESP using corresponding algorithms to TH2.
27. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 7: Judgment #3
The NUT forwards an Echo Reply.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 11: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 13: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 15: Judgment #3
The NUT forwards an Echo Reply.

Step 17: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 19: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 21: Judgment #2
The NUT transmits an IKE_AUTH request including an ID payload which contains the value specified as above table.

Step 23: Judgment #3
The NUT forwards an Echo Reply.

Step 25: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.10 Certificate Payload

Test IKEv2.SGW.I.1.1.10.1: Sending CERT Payload

Purpose:

To verify an IKEv2 device handles CERTREQ payload and transmits CERT payload properly.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Remote	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr, CERTREQ)
  |                  | (Packet #1)
  |                  |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, CERT, AUTH, SAi2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See below

Packet #1: IKE_SA_INIT response

IPv6 Header	Same as the Common Packet #2	
UDP Header	Same as the Common Packet #2	
IKEv2 Header	Same as the Common Packet #2	
SA Payload	Same as the Common Packet #2	
KE Payload	Same as the Common Packet #2	
Nr Payload	Next Payload	38 (CERTREQ)
	Other fields are same as the Common Packet #2	
CERTREQ Payload	See below	

CERTREQ Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Authority	any

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT request from the NUT, TN1 responds with an IKE_SA_INIT response with a CERTREQ payload to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request with a CERT payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding and the NUT’s certificate as Certificate Data.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.10.2: Sending CERTREQ Payload

Purpose:

To verify an IKEv2 device transmits CERTREQ payload and handles CERT payload properly.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #2

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.10.3: RSA Digital Signature

Purpose:

To verify an IKEv2 device authenticates the corresponding node by RSA Digital Signature.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1             NUT            TN1            TH2
(Host)          (SGW)          (SGW)          (Host)
  |               |              |              |
  |               |------------->|              | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               |              |              | (Judgment #1)
  |               |<-------------|              | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |               |              |              | (Packet #1)
  |               |              |              |
  |               |------------->|              | IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH,
  |               |              |              |                            SAi2, TSi, TSr})
  |               |              |              | (Judgment #2)
  |               |<-------------|              | IKE_AUTH response (HDR, SK {IDr, CERT, AUTH,
  |               |              |              |                             SAr2, TSi, TSr})
  |               |              |              | (Packet #2)
  |               |              |              |
  |<--------------+==============+--------------| IPsec {Echo Request}
  |               |              |              | (Packet #3) (Judgment #3)
  |---------------+==============+------------->| IPsec {Echo Reply}
  |               |              |              | (Packet #4) (Judgment #4)
  |               |              |              |
  V               V              V              V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #19

Packet #2: IKE_AUTH response

IPv6 Header	Same as Common Packet #6
UDP Header	Same as Common Packet #6

IKEv2 Header	Same as Common Packet #6	
E Payload	Same as Common Packet #6	
IDr Payload	Next Payload	37 (CERT)
	Other fields are same as the Common Packet #6	
CERT Payload	See below	
AUTH Payload	Same as Common Packet #6	
SA Payload	Same as Common Packet #6	
TSi Payload	Same as Common Packet #6	
TSr Payload	Same as Common Packet #6	

CERT Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Data	TN1's X.509 Certificate

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response including IDr payload as describe above to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.10.4: HEX string PSK

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 2.15

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Key Value
Remote	0xabadcafeabadcafeabadcafeabadcafe (128 bit binary string)

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                    TN1
(SGW)                  (SGW)
  |                      |
  |--------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                      | (Judgment #1)
  |<---------------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                      | (Packet #1)
  |                      |
  |--------------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                      | (Judgment #2)
  |                      |
  V                      V
```

Packet #1	See Common Packet #2

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Group 1.11 Invalid values

Test IKEv2.SGW.I.1.1.11.1: Non zero RESERVED fields in IKE_SA_INIT response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #2 All RESERVED fields are set to one.

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response whose RESERVED fields are set to one to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.11.2: Non zero RESERVED fields in IKE_AUTH response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |-------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #1)
  |                |<--------------|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #1)
  |                |                |                |
  |                |-------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |                |                |                     SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |<--------------|                | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                |                |                |                      SAr2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |                |                |
  |<--------------+===============+--------------| IPsec {Echo Request}
  |                |                |                | (Packet #3) (Judgment #3)
  |--------------+===============+-------------->| IPsec {Echo Reply}
  |                |                |                | (Packet #4) (Judgment #4)
  |                |                |                |
  V                V                V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6 All RESERVED fields are set to one.
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH

response whose RESERVED fields are set to one to the NUT

6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.11.3: Version bit is set

Purpose:

To verify an IKEv2 device ignores the content of Version bit in IKE messages.

References:

- [RFC 4306] - Sections 3.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #2 Version bit is set to one.

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response whose Version bit is set to one to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.11.4: Unrecognized Notify Message Type of Error

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type intended for reporting error.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                  TN1
 (SGW)                (SGW)
   |                    |
   |------------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                    | (Judgment #1)
   |<-------------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, N)
   |                    | (Packet #1)
   |                    |
   |-------X            | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                    | (Judgment #2)
   |                    |
   V                    V

N: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #2	
UDP Header	All fields are same as Common Packet #2	
IKEv2 Header	All fields are same as Common Packet #2	
SA Payload	All fields are same as Common Packet #2	
KE Payload	All fields are same as Common Packet #2	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #2	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	16383

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response with a Notify payload of unrecognized Notify Message Type value.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT never transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.1.11.5: Unrecognized Notify Message Type of Status

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type intended for reporting status.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
  (SGW)           (SGW)
    |               |
    |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Judgment #1)
    |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, N)
    |               | (Packet #1)
    |               |
    |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    V               V

N: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #2	
UDP Header	All fields are same as Common Packet #2	
IKEv2 Header	All fields are same as Common Packet #2	
SA Payload	All fields are same as Common Packet #2	
KE Payload	All fields are same as Common Packet #2	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #2	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	65535

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A.
7. TN1 responds with an IKE_SA_INIT response with a Notify payload of unrecognized Notify Message Type value.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Group 2. The CREATE_CHILD_SA Exchange

Group 2.1. Header and Payload Formats

Test IKEv2.SGW.I.1.2.1.1: Sending CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device transmits CREATE_CHILD_SA request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<--------------|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<--------------|                  | IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
 ...                ...                ...                ...
  |                  |                  |                  |                        ---
  |<-------------+==============+-------------| IPsec {Echo Request }      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+==============+------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                        ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link B.

Part B: Encrypted Payload Format (BASIC)

12. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
13. Observe the messages transmitted on Link B.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link B.
16. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
17. TH2 transmits an Echo Request to TH1.
18. Observe the messages transmitted on Link A.

19. TH1 transmits an Echo Reply to TH2.
20. Observe the messages transmitted on Link B.
21. Repeat Steps 6 through 9 until lifetime of SA is expired.
22. Observe the messages transmitted on Link B.

Part C: Notify Payload (REKEY_SA) Format (BASIC)

23. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
24. Observe the messages transmitted on Link B.
25. TN1 responds with an IKE_SA_INIT response to the NUT.
26. Observe the messages transmitted on Link B.
27. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
28. TH2 transmits an Echo Request to TH1.
29. Observe the messages transmitted on Link A.
30. TH1 transmits an Echo Reply to TH2.
31. Observe the messages transmitted on Link B.
32. Repeat Steps 6 through 9 until lifetime of SA is expired.
33. Observe the messages transmitted on Link B.

Part D: SA Payload Format (BASIC)

34. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
35. Observe the messages transmitted on Link B.
36. TN1 responds with an IKE_SA_INIT response to the NUT.
37. Observe the messages transmitted on Link B.
38. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
39. TH2 transmits an Echo Request to TH1.
40. Observe the messages transmitted on Link A.
41. TH1 transmits an Echo Reply to TH2.
42. Observe the messages transmitted on Link B.
43. Repeat Steps 6 through 9 until lifetime of SA is expired.
44. Observe the messages transmitted on Link B.

Part E: Nonce Payload Format (BASIC)

45. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
46. Observe the messages transmitted on Link B.
47. TN1 responds with an IKE_SA_INIT response to the NUT.
48. Observe the messages transmitted on Link B.
49. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
50. TH2 transmits an Echo Request to TH1.
51. Observe the messages transmitted on Link A.
52. TH1 transmits an Echo Reply to TH2.
53. Observe the messages transmitted on Link B.
54. Repeat Steps 6 through 9 until lifetime of SA is expired.
55. Observe the messages transmitted on Link B.

Part F: TSi Payload Format (BASIC)

56. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
57. Observe the messages transmitted on Link B.
58. TN1 responds with an IKE_SA_INIT response to the NUT.
59. Observe the messages transmitted on Link B.
60. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH

response to the NUT

61. TH2 transmits an Echo Request to TH1.
62. Observe the messages transmitted on Link A.
63. TH1 transmits an Echo Reply to TH2.
64. Observe the messages transmitted on Link B.
65. Repeat Steps 6 through 9 until lifetime of SA is expired.
66. Observe the messages transmitted on Link B.

Part G: TSr Payload Format (BASIC)

67. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
68. Observe the messages transmitted on Link B.
69. TN1 responds with an IKE_SA_INIT response to the NUT.
70. Observe the messages transmitted on Link B.
71. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
72. TH2 transmits an Echo Request to TH1.
73. Observe the messages transmitted on Link A.
74. TH1 transmits an Echo Reply to TH2.
75. Observe the messages transmitted on Link B.
76. Repeat Steps 6 through 9 until lifetime of SA is expired.
77. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 119 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to CREATE_CHILD_SA (36).
- A Flags field set to (00010000)2 = (16)10.
- A Message ID field set to the value incremented the previous IKE message's Message ID by one.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 13: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 15: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 18: Judgment #3
The NUT forwards an Echo Request.

Step 20: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 22: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 120 Encrypted payload

- A Next Payload field set to N Payload (41).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 24: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 26: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 29: Judgment #3
The NUT forwards an Echo Request.

Step 31: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 33: Judgment #5

The NUT transmits a CREATE_CHILD_SA request including properly formatted Notify Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 121 Notify Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 12 bytes for this REKEY_SA.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to the size of CHILD_SA Inbound SPI value to be rekeyed. It is 4 bytes for ESP.
- A Notify Message Type field set to REKEY_SA (16393).
- A Security Parameter Index field set to SPI value to be rekeyed.
- A Notification Data field is empty.

Part D

Step 35: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 37: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 40: Judgment #3
The NUT forwards an Echo Request.

Step 42: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 44: Judgment #5

```
                                  1                   2                   3
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
             ! Next     44    !0!      0      ! Length          40            !              |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---           |
             !       0        !        0      ! Length          36            !  |           |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
             ! Number     1   ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3  !  |           |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
             ! SPI value                                                      !  |           |
         --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
           | !       3        !        0      ! Length          8             !  |           |
Transform  | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |SA Payload
           | ! Type  1  (EN) !        0       ! Transform ID     3      (3DES) !  | Proposal  |
         --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
           | !       3        !        0      ! Length          8             !  |           |
Transform  | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
           | ! Type  3  (IN) !        0       ! Transform ID     2      (SHA1) !  |           |
         --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
           | !       0        !        0      ! Length          8             !  |           |
Transform  | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
           | ! Type  5  (ESN)!        0       ! Transform ID     0      (No)   !  |           |
         --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 122 SA Payload contents

The NUT transmits a CREATE_CHILD_SA request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 123 SA Payload format

- A Next Payload field set to Ni Payload (40).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 124 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1 if this structure is the first proposal, otherwise set to 1 greater thatn the previous proposal.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 125 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.

- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part E

Step 46: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 48: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 51: Judgment #3

The NUT forwards an Echo Request.

Step 53: Judgment #4

The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 55: Judgment #5

The NUT transmits a CREATE_CHILD_SA request including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 126 Nonce Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Nonce Data field set to random data generated by the transmitting entity.
- The size of the Nonce must between 16 and 256 octets.

Part F

Step 57: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 59: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 62: Judgment #3
The NUT forwards an Echo Request.

Step 64: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 66: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 127 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 128 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- A Ending Address field set to greater thatn or equal to Prefix B.

Part G

Step 68: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 70: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 73: Judgment #3
The NUT forwards an Echo Request.

Step 75: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 77: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 129 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 130 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix Y.
- An Ending Address field set to less than or equal to Prefix Y.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TH2 can send Echo Reply to TH1 instead of sending Echo Request.
- The implementation may use different SA lifetimes by the implementation policy. In

that case, the tester must change the expiration time to wait CREATE_CHILD_SA request.

- CREATE_CHILD_SA request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(REKEY_SA)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, Ni, [KEi], TSi, TSr
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- Each of transforms can be located in the any order.

- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

Group 2.2. Use of Retransmission Timers

Test IKEv2.SGW.I.1.2.2.1: Retransmissions of CREATE_CHILD_SA requests

Purpose:

To verify an IKEv2 device retransmits CREATE_CHILD_SA request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |-------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<--------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |-------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<--------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
 ...                ...                ...                ...
  |                  |                  |                  |                           ---
  |<--------------+===============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+-------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                           ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |-------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |                  |                  * wait for the event of a timeout
  |                  |                  |                  |
  |                  |-------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Judgment #6)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH1 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link B.
12. TN1 waits for the event of a timeout on NUT.
13. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 13: Judgment #6
The NUT retransmits a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different lifetime of SA.
- Each NUT has the different retransmission timers.

Test IKEv2.SGW.I.1.2.2.2: Stop of retransmission of CREATE_CHILD_SA requests

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |-------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #1)
  |                |<--------------|                | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #1)
  |                |                |                |
  |                |-------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |<--------------|                | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |                |                |
 ...              ...              ...              ...
  |                |                |                |                          ---
  |<--------------+===============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                |                |                | (Packet #3) (Judgment #3) | until lifetime of SA
  |---------------+===============+------------->| IPsec {Echo Reply}        | is expired
  |                |                |                | (Packet #4) (Judgment #4) |
  |                |                |                |                          ---
 ...              ...              ...              ...
  |                |                |                |
  |                |-------------->|                | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |                |                | (Judgment #5)
  |                |                |                |
  |                |                |                * wait for the event of a timeout
  |                |                |                |
  |                |-------------->|                | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |                |                | (Judgment #6)
  |                |                |                |
  |                |<--------------|                | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                |                |                | (Packet #5)
  |                |                |                |
  |                |                |                * wait for the event of a timeout
  |                |                |                |
  |                |-------X        |                | never send CREATE_CHILD_SA request (HDR, SK {N, SA, Ni,
  |                |                |                |                                      TSi, TSr})
  |                |                |                | (Judgment #7)
  |                |                |                |
  V                V                V                V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Request to TH2.
9. Observe the messages transmitted on Link B.

10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link B.
12. TN1 waits for the event of a timeout on NUT.
13. Observe the messages transmitted on Link B.
14. TN1 responds with a CREATE_CHILD_SA response to the NUT.
15. TN1 waits for the event of a timeout on NUT.
16. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 13: Judgment #6
The NUT retransmits a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request's Message ID value in IKE Header.

Step 16: Judgment #7
The NUT stops the retransmissions of a CREATE_CHILD_SA request which has the same Message ID value as the previous CREATE_CHILD_SA request's Message ID value in IKE Header.

Possible Problems:

- Each NUT has the different lifetime of SA.
- Each NUT has the different retransmission timers.

Group 2.3. Rekeying CHILD_SA Using a CREATE_CHILD_SA exchange

Test IKEv2.SGW.I.1.2.3.1: Close the replaced CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------->|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<--------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------->|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<--------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                          ---
  |<-------------+===============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+------------->| IPsec {Echo Reply}        | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4) |
  |                   |                   |                   |                          ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |------------->|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Judgment #5)
  |                   |                   |                   |
  |                   |<--------------|                   | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                   |                   |                   | (Packet #5)
  |                   |                   |                   |
  |                   |------------->|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Judgment #6)
  V                   V                   V                   V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.2: Use of the new CHILD_SA

Purpose:

To verify an IKEv2 device properly rekeys CHILD_SA

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                  NUT                  TN1                  TH2
(Host)               (SGW)                (SGW)                (Host)
  |                    |                    |                    |
  |                    |------------------->|                    | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    |                    |                    | (Judgment #1)
  |                    |<-------------------|                    | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                    |                    |                    | (Packet #1)
  |                    |                    |                    |
  |                    |------------------->|                    | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                    |                    |                    | (Judgment #2)
  |                    |<-------------------|                    | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                    |                    |                    | (Packet #2)
  |                    |                    |                    |
 ...                  ...                  ...                  ...
  |                    |                    |                    |                               ---
  |<-------------------+====================+--------------------| IPsec {Echo Request}        | repeat Echo exchange
  |                    |                    |                    | (Packet #3) (Judgment #3)   | until lifetime of SA
  |--------------------+====================+------------------->| IPsec {Echo Reply}          | is expired
  |                    |                    |                    | (Packet #4) (Judgment #4)   |
  |                    |                    |                    |                               ---
 ...                  ...                  ...                  ...
  |                    |                    |                    |
  |                    |------------------->|                    | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                    |                    |                    | (Judgment #5)
  |                    |                    |                    |
  |                    |<-------------------|                    | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                    |                    |                    | (Packet #5)
  |                    |                    |                    |
  |                    |------------------->|                    | INFORMATIONAL request (HDR, SK {D})
  |                    |                    |                    | (Judgment #6)
  |                    |<-------------------|                    | INFORMATIONAL response (HDR, SK {D})
  |                    |                    |                    | (Packet #6)
  |                    |                    |                    |
  |<-------------------+====================+--------------------| IPsec {Echo Request}
  |                    |                    |                    | (Packet #7) (Judgment #7)
  |--------------------+====================+------------------->| IPsec {Echo Request}
  |                    |                    |                    | (Packet #8) (Judgment #8)
  |                    |                    |                    |
  V                    V                    V                    V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16
Packet #6	See below
Packet #7	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 11.
Packet #8	See Common Packet #25

Packet #6: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any

	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response with a Delete payload to the NUT.
15. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
16. Observe the messages transmitted on Link B.
17. TH1 transmits an Echo Response to the TH2.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Step 16: Judgment #7
The NUT forwards an Echo Request to the TH1.

Step 18: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.3: Lifetime of CHILD_SA expires

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of CHILD_SAs.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1           NUT            TN1            TH2
(Host)        (SGW)          (SGW)          (Host)
  |             |              |              |
  |             |------------->|              | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |             |              |              | (Judgment #1)
  |             |<-------------|              | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |             |              |              | (Packet #1)
  |             |              |              |
  |             |------------->|              | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |             |              |              | (Judgment #2)
  |             |<-------------|              | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |             |              |              | (Packet #2)
  |             |              |              |
  |<------------+==============+--------------| IPsec {Echo Request}
  |             |              |              | (Packet #3) (Judgment #3)
  |-------------+==============+------------->| IPsec {Echo Reply}
  |             |              |              | (Packet #4) (Judgment #4)
  |             |              |              |
  |             |              |              * wait for the event of a timeout of CHILD_SA
  |             |              |              |
  |       X-----+==============+--------------| IPsec {Echo Request}
  |             |              |              | (Packet #5) (Judgment #5)
  |-------------+=======X      |              | IPsec {Echo Reply}
  |             |              |              | (Packet #6) (Judgment #6)
  |             |              |              |
  V             V              V              V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #21

Packet #6	See Common Packet #25

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Request to TH2.
9. Observe the messages transmitted on Link B.
10. TN1 waits for the event of a timeout on the NUT.
11. After timeout of CHILD_SA on the NUT, TH2 transmits an Echo Request to the TH1.
12. Observe the messages transmitted on Link A.
13. TH1 transmits an Echo Request to TH2.
14. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 12: Judgment #5
The NUT does not forward an Echo Request.

Step 14: Judgment #6
The NUT does not forward an Echo Reply with IPsec ESP using already expired CHILD_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.4: Sending Multiple Transform

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	CREATE_CHILD_SA exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN** **ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT             TN1             TH2
(Host)           (SGW)           (SGW)           (Host)
  |                |               |               |
  |                |-------------->|               | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |               |               | (Judgment #1)
  |                |<--------------|               | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |               |               | (Packet #1)
  |                |               |               |
  |                |-------------->|               | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |               |               | (Judgment #2)
  |                |<--------------|               | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |               |               | (Packet #2)
  |                |               |               |
 ...              ...             ...             ...
  |                |               |               |                         ---
  |<---------------+===============+---------------| IPsec {Echo Request}      | repeat Echo exchange
  |                |               |               | (Packet #3) (Judgment #3) | until lifetime of SA
  |----------------+===============+-------------->| IPsec {Echo Reply}        | is expired
  |                |               |               | (Packet #4) (Judgment #4) |
  |                |               |               |                         ---
 ...              ...             ...             ...
  |                |               |               |
  |                |-------------->|               | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |               |               | (Judgment #5)
  |                |               |               |
  V                V               V               V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packets #6
Packet #3	See Common Packets #21
Packet #4	See Common Packet #25

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link B.

Part B: Multiple Integrity Algorithms (ADVANCED)

12. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
13. Observe the messages transmitted on Link B.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link B.
16. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
17. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP

using the first negotiated algorithms to NUT.

18. Observe the messages transmitted on Link A.
19. TH1 transmits an Echo Reply to TH2.
20. Observe the messages transmitted on Link B.
21. Repeat Steps 17 through 20 until lifetime of SA is expired.
22. Observe the messages transmitted on Link B.

Part C: Multiple Extended Sequecnce Numbers (ADVANCED)

23. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
24. Observe the messages transmitted on Link B.
25. TN1 responds with an IKE_SA_INIT response to the NUT.
26. Observe the messages transmitted on Link B.
27. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
28. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
29. Observe the messages transmitted on Link A.
30. TH1 transmits an Echo Reply to TH2.
31. Observe the messages transmitted on Link B.
32. Repeat Steps 28 through 31 until lifetime of SA is expired.
33. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Part B

Step 13: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 15: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 18: Judgment #3
The NUT forwards an Echo Request.

Step 20: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 22: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Part C

Step 24: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 26: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 29: Judgment #3
The NUT forwards an Echo Request.

Step 31: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 33: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96”, “No Extended Sequence Numbers” and “Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.5: Sending Multiple Proposal

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple proposals to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration:

	CREATE_CHILD_SA exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	**ESP**	**ENCR_AES_CBC**	**AUTH_AES_XCBC_96**	**ESN**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                  NUT                  TN1                  TH2
(Host)               (SGW)                (SGW)                (Host)
  |                    |                    |                    |
  |                    |------------->|                    | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    |                    |                    | (Judgment #1)
  |                    |<--------------|                    | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                    |                    |                    | (Packet #1)
  |                    |                    |                    |
  |                    |------------->|                    | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                    |                    |                    | (Judgment #2)
  |                    |<--------------|                    | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                    |                    |                    | (Packet #2)
  |                    |                    |                    |
 ...                  ...                  ...                  ...
  |                    |                    |                    |                         ---
  |<-------------+==============+-------------| IPsec {Echo Request}       | repeat Echo exchange
  |                    |                    |                    | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+==============+------------->| IPsec {Echo Reply}         | is expired
  |                    |                    |                    | (Packet #4) (Judgment #4) |
  |                    |                    |                    |                         ---
 ...                  ...                  ...                  ...
  |                    |                    |                    |
  |                    |------------->|                    | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                    |                    |                    | (Judgment #5)
  |                    |                    |                    |
  V                    V                    V                    V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packets #6
Packet #3	See Common Packets #21
Packet #4	See Common Packet #25

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” in SA Proposal #1 (ESP) and then “ENCR_AES_CBC”, “AUTH_AES_XCBC_96” and “Extended Sequence Numbers” in SA Proposal #2 (ESP) as accepted algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.6: Rekying Failure

Purpose:

To verify an IKEv2 device properly handles rekeying failure.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |--------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #1)
  |                |<---------------|                | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #1)
  |                |                |                |
  |                |--------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |<---------------|                | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |                |                |
 ...              ...              ...              ...
  |                |                |                |                          ---
  |<---------------+================+----------------| IPsec {Echo Request}      | repeat Echo exchange
  |                |                |                | (Packet #3) (Judgment #3) | until lifetime of SA
  |----------------+================+--------------->| IPsec {Echo Reply}        | is expired
  |                |                |                | (Packet #4) (Judgment #4) |
  |                |                |                |                          ---
 ...              ...              ...              ...
  |                |                |                |
  |                |--------------->|                | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |                |                | (Judgment #5)
  |                |                |                |
  |                |<---------------|                | CREATE_CHILD_SA response (HDR, SK(N(NO_PROPOSAL_CHOSEN))
  |                |                |                | (Packet #5)
  |                |                |                |
  |                |--------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Judgment #6)
  |                |<---------------|                | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Packet #6)
  |                |                |                |
  |                |--------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                |                | (Judgment #7)
  V                V                V                V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See Common Packet #2

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.

11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 reject the NUT's proposal and responds with a CREATE_CHILD_SA response with a Notify of type NO_PROPOSAL_CHOSEN.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 15: Judgment #7
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.7: Perfect Forward Secrecy

Purpose:

To verify an IKEv2 device properly rekeys CHILD_SA when Perfect Forward Secrecy enables.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds. Enable PFS.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------->|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<-------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------->|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<-------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                          ---
  |<-------------+===============+--------------| IPsec {Echo Request}       | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+------------->| IPsec {Echo Reply}         | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4) |
  |                   |                   |                   |                          ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |------------->|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Judgment #5)
  |                   |                   |                   |
  |                   |<-------------|                   | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                   |                   |                   | (Packet #5)
  |                   |                   |                   |
  |                   |------------->|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Judgment #6)
  |                   |<-------------|                   | INFORMATIONAL response (HDR, SK {D})
  |                   |                   |                   | (Packet #6)
  |                   |                   |                   |
  |<-------------+===============+--------------| IPsec {Echo Request}
  |                   |                   |                   | (Packet #7) (Judgment #7)
  |--------------+===============+------------->| IPsec {Echo Request}
  |                   |                   |                   | (Packet #8) (Judgment #8)
  |                   |                   |                   |
  V                   V                   V                   V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below
Packet #7	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 11.
Packet #8	See Common Packet #25

Packet #5: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #16
UDP Header	Same as the Common Packet #16
IKEv2 Header	Same as the Common Packet #16
E Payload	Same as the Common Packet #16
N Payload	Same as the Common Packet #16

N	Same as the Common Packet #16	
SA	Same as the Common Packet #16	
Nr	Next Payload	34 (KE)
KEr	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	136
	DH Group #	2
	Reserved	0
	Key Exchange Data	any
TSi	Same as the Common Packet #16	
TSr	Same as the Common Packet #16	

Packet #6: INFORMATIONAL response

IPv6 Header	Same as the Common Packet #18	
UDP Header	Same as the Common Packet #18	
IKEv2 Header	Same as the Common Packet #18	
E Payload	Other fields are same as the Common Packet #18	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response with a Delete payload to the NUT.
15. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the newly negotiated algorithms to NUT.
16. Observe the messages transmitted on Link B.
17. TH1 transmits an Echo Response to the TH2.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Step 16: Judgment #7
The NUT forwards an Echo Request to the TH1.

Step 18: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using the second negotitated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.3.8: Use of the old CHILD_SA

Purpose:

To verify an IKEv2 device properly handles new CHILD_SA and old CHILD_SA

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<---------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<---------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
 ...                ...                ...                ...
  |                  |                  |                  |                           ---
  |<---------------+================+---------------| IPsec {Echo Request}      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |---------------+================+--------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                           ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |--------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |<---------------|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                  |                  |                  | (Packet #5)
  |                  |                  |                  |
  |<---------------+================+---------------| IPsec {Echo Request} (old CHILD_SA)
  |                  |                  |                  | (Packet #6) (Judgment #6)
  |---------------+================+--------------->| IPsec {Echo Request} (old CHILD_SA)
  |                  |                  |                  | (Packet #7) (Judgment #7)
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16
Packet #6	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 5.
Packet #7	See Common Packet #25

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. TH2 transmits an Echo Request to the TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms again.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Response to the TH2.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5

The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 14: Judgment #6
The NUT forwards an Echo Request to the TH1.

Step 16: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.4. Rekeying IKE_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.SGW.I.1.2.4.1: Close the replaced IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<------------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<------------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                     ---
  |<------------------+===================+-------------------| IPsec {Echo Request}     | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |-------------------+===================+------------------>| IPsec {Echo Reply}       | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4) |
  |                   |                   |                   |                     ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |------------------>|                   | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                   |                   |                   | (Judgment #5)
  |                   |<------------------|                   | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                   |                   |                   | (Packet #5)
  |                   |                   |                   |
  |                   |------------------>|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Judgment #6)
  |                   |<------------------|                   | INFROMATIONAL response (HDR, SK { })
  |                   |                   |                   | (Packet #6)
  |                   |                   |                   |
  |<------------------+===================+-------------------| IPsec {Echo Request}
  |                   |                   |                   | (Packet #7) (Judgment #7)
  |-------------------+===================+------------------>| IPsec {Echo Reply}
  |                   |                   |                   | (Packet #8) (Judgment #8)
  |                   |                   |                   |
  V                   V                   V                   V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #12
Packet #6	See Common Packet #18
Packet #7	See Common Packet #21
Packet #8	See Common Packet #25

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.

9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response to close the replaced IKE_SA.
15. TH2 transmits an Echo Request to TH1. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.
16. Observe the messages transmitted on Link A.
17. TH1 transmits an Echo Reply to TH2.
18. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 16: Judgment #7
The NUT forwards an Echo Request.

Step 18: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.2: Use of the new IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1             NUT             TN1             TH2
(Host)          (SGW)           (SGW)           (Host)
  |               |               |               |
  |               |-------------->|               | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               |               |               | (Judgment #1)
  |               |<--------------|               | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |               |               |               | (Packet #1)
  |               |               |               |
  |               |-------------->|               | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               |               |               | (Judgment #2)
  |               |<--------------|               | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               |               |               | (Packet #2)
  |               |               |               |
 ...             ...             ...             ...
  |               |               |               |                            ---
  |<--------------+===============+---------------| IPsec {Echo Request}       | repeat Echo exchange
  |               |               |               | (Packet #3) (Judgment #3)  | until lifetime of SA
  |---------------+===============+-------------->| IPsec {Echo Reply}         | is expired
  |               |               |               | (Packet #4) (Judgment #4)  |
  |               |               |               |                            ---
 ...             ...             ...             ...
  |               |               |               |
  |               |-------------->|               | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |               |               |               | (Judgment #5)
  |               |<--------------|               | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |               |               |               | (Packet #5)
  |               |               |               |
  |               |-------------->|               | INFORMATIONAL request (HDR, SK {D})
  |               |               |               | (Judgment #6)
  |               |<--------------|               | INFROMATIONAL response (HDR, SK { })
  |               |               |               | (Packet #6)
  |               |               |               |
  |               |<--------------|               | INFORMATIONAL request (HDR, SK {})
  |               |               |               | (Packet #7)
  |               |-------------->|               | INFORMATIONAL response (HDR, SK {})
  |               |               |               | (Judgment #7)
  |               |               |               |
  V               V               V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #12
Packet #6	See Common Packet #18
Packet #7	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.

11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close the replaced IKE_SA.
15. TN1 transmits an INFORMATIONAL request with no payloads cryptographically protected by new IKE_SA.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA's SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 16: Judgment #7
The NUT responds with an INFORMATIONAL response with no payloads cryptographically protected by the new IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.3: Lifetime of IKE_SA expires

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |<---------------| INFORMATION Request (HDR, SK {})
  |                | (Packet #3)
  |--------------->| INFORMATIONAL response (HDR, SK {})
  |                | (Judgment #3)
  |                |
  |                * wait for the event of a timeout of IKE_SA
  |                |
  |<---------------| INFORMATION Request (HDR, SK {})
  |                | (Packet #4)
  |--------X       | INFORMATIONAL response (HDR, SK {})
  |                | (Judgment #4)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #17
Packet #4	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
7. Observe the messages transmitted on Link B.
8. TN1 waits for the event of a timeout on the NUT.
9. After timeout of CHILD_SA on the NUT, TN1 transmits an INFORMATIONAL request with no payloads using already expired IKE_SA.
10. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT responds with an INFORMATIONAL response with no payloads.

Step 10: Judgment #4
The NUT does not respond with an INFORMATIONAL response with no payloads using already expired IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.4: Sending Multiple Transform

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple transforms to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.

	CREATE_CHILD_SA exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_HMAC_SHA1** **PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 2** **Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1               NUT               TN1               TH2
(Host)            (SGW)             (SGW)             (Host)
  |                 |                 |                 |
  |                 |---------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 |                 |                 | (Judgment #1)
  |                 |<----------------|                 | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                 |                 |                 | (Packet #1)
  |                 |                 |                 |
  |                 |---------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 |                 |                 | (Judgment #2)
  |                 |<----------------|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 |                 |                 | (Packet #2)
  |                 |                 |                 |
 ...               ...               ...               ...
  |                 |                 |                 |                             ---
  |<----------------+=================+-----------------| IPsec {Echo Request}        | repeat Echo exchange
  |                 |                 |                 | (Packet #3) (Judgment #3)   | until lifetime of SA
  |-----------------+=================+---------------->| IPsec {Echo Reply}          | is expired
  |                 |                 |                 | (Packet #4) (Judgment #4)   |
  |                 |                 |                 |                             ---
 ...               ...               ...               ...
  |                 |                 |                 |
  |                 |---------------->|                 | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                 |                 |                 | (Judgment #5)
  V                 V                 V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo-Random Functions (ADVANCED)

12. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an IKE_SA_INIT response to the NUT.
15. Observe the messages transmitted on Link A.
16. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
17. TH2 transmits an Echo Request to TH1.
18. Observe the messages transmitted on Link B.
19. TH1 transmits an Echo Reply to TH2.
20. Observe the messages transmitted on Link A.
21. Repeat Steps 17 through 20 until lifetime of SA is expired.

22. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (ADVANCED)

23. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
24. Observe the messages transmitted on Link A.
25. TN1 responds with an IKE_SA_INIT response to the NUT.
26. Observe the messages transmitted on Link A.
27. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
28. TH2 transmits an Echo Request to TH1.
29. Observe the messages transmitted on Link B.
30. TH1 transmits an Echo Reply to TH2.
31. Observe the messages transmitted on Link A.
32. Repeat Steps 28 through 31 until lifetime of SA is expired.
33. Observe the messages transmitted on Link A.

Part D: Multiple D-H Groups (ADVANCED)

34. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
35. Observe the messages transmitted on Link A.
36. TN1 responds with an IKE_SA_INIT response to the NUT.
37. Observe the messages transmitted on Link A.
38. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
39. TH2 transmits an Echo Request to TH1.
40. Observe the messages transmitted on Link B.
41. TH1 transmits an Echo Reply to TH2.
42. Observe the messages transmitted on Link A.
43. Repeat Steps 39 through 42 until lifetime of SA is expired.
44. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Part B

Step 13: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 15: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 18: Judgment #3
The NUT forwards an Echo Request.

Step 20: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 22: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “PRF_AES128_CBC”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Part C

Step 24: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 26: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 29: Judgment #3
The NUT forwards an Echo Request.

Step 31: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 33: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “AUTH_AES_XCBC_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Part D

Step 35: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 37: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 40: Judgment #3
The NUT forwards an Echo Request.

Step 42: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 44: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”,“PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96”, “D-H group 2” and “D-H group 14” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.5: Sending Multiple Proposal

Purpose:

To verify an IKEv2 device properly transmits CREATE_CHILD_SA request with multiple proposal to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.

	CREATE_CHILD_SA exchanges Algorithms					
	Proposal	**Protocol ID**	**Encryption**	**PRF**	**Integrity**	**D-H Group**
Part A	Proposal #1	IKE	ENCR_ 3DES	PRF_ HMAC_SHA1	AUTH_ HMAC_SHA1_96	Group 2
	Proposal #2	**IKE**	**ENCR_ AES_CBC**	**PRF_ AES128_CBC**	**AUTH_ AES_XCBC_96**	**Group 14**

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |-------------->|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<--------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |-------------->|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<--------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                          ---
  |<--------------+===============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+-------------->| IPsec {Echo Reply}        | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4) |
  |                   |                   |                   |                          ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |-------------->|                   | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                   |                   |                   | (Judgment #5)
  V                   V                   V                   V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: Multiple Encryption Algorithms (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” in SA Proposal #1 (ESP) and “ENCR_AES_CBC”, “PRF_AES128_CBC”, “AUTH_AES_XCBC_96” and “D-H group 14” in SA Proposal #2 (ESP) as proposed algorithms.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.6: Use of the old IKE_SA

Purpose:

To verify an IKEv2 device properly handles new IKE_SA and old IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1             NUT             TN1             TH2
(Host)          (SGW)           (SGW)           (Host)
  |               |               |               |
  |               |-------------->|               | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               |               |               | (Judgment #1)
  |               |<--------------|               | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |               |               |               | (Packet #1)
  |               |               |               |
  |               |-------------->|               | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               |               |               | (Judgment #2)
  |               |<--------------|               | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               |               |               | (Packet #2)
  |               |               |               |
 ...             ...             ...             ...
  |               |               |               |                           ---
  |<--------------+===============+---------------| IPsec {Echo Request}      | repeat Echo exchange
  |               |               |               | (Packet #3) (Judgment #3) | until lifetime of SA
  |---------------+===============+-------------->| IPsec {Echo Reply}        | is expired
  |               |               |               | (Packet #4) (Judgment #4) |
  |               |               |               |                           ---
 ...             ...             ...             ...
  |               |               |               |
  |               |-------------->|               | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |               |               |               | (Judgment #5)
  |               |               |               |
  |               |<--------------|               | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |               |               |               | (Packet #5)
  |               |               |               |
  |               |<--------------|               | INFORMATIONAL request (HDR, SK {})
  |               |               |               | (Packet #6)
  |               |-------------->|               | INFORMATIONAL response (HDR, SK {})
  |               |               |               | (Judgment #6)
  V               V               V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #12
Packet #6	See Common Packet #17 (Use old IKE_SA)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request to rekey IKE_SA from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is encrypted by the old IKE_SA.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 14: Judgment #6

The NUT transmits an INFORMATIONAL response with no payload. The message is encrypted by the old IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.4.7: Changing PRFs when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
  Configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA Rekeying Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<---------------|                  | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<---------------|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
 ...                ...                ...                ...
  |                  |                  |                  |                          ---
  |<-----------------+==================+------------------| IPsec {Echo Request}      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |------------------+==================+----------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                          ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |--------------->|                  | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                  |                  |                  | (Judgment #5)
  |                  |<---------------|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                  |                  |                  | (Packet #5)
  |                  |                  |                  |
  |                  |--------------->|                  | INFORMATIONAL request (HDR, SK {D})
  |                  |                  |                  | (Judgment #6)
  |                  |<---------------|                  | INFROMATIONAL response (HDR, SK { })
  |                  |                  |                  | (Packet #6)
  |                  |                  |                  |
  |                  |<---------------|                  | INFORMATIONAL request (HDR, SK {})
  |                  |                  |                  | (Packet #7)
  |                  |--------------->|                  | INFORMATIONAL response (HDR, SK {})
  |                  |                  |                  | (Judgment #7)
  |                  |                  |                  |
  V                  V                  V                  V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See Common Packet #18
Packet #7	See Common Packet #17

Packet #5: CREATE_CHILD_SA response

Packet #5 is same as Common Packet #12 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	14 (2048 MODP Group)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link B.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close the replaced IKE_SA.
15. TN1 transmits an INFORMATIONAL request with no payloads cryptographically protected by new IKE_SA.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload to close the replaced IKE_SA.

Step 16: Judgment #7

The NUT responds with an INFORMATIONAL response with no payloads cryptographically protected by the new IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.5. Creating New CHILD_SAs with the CREATE_CHILD_SA Exchanges

Test IKEv2.SGW.I.1.2.5.1: Create new CHILD_SA by sending CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to generate new CHILD_SAs.

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3
- [RFC 4718] - Sections 4.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT             TN1
(SGW)           (SGW)
  |               |
  |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               | (Judgment #1)
  |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |               | (Packet #1)
  |               |
  |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               | (Judgment #2)
  |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               | (Packet #2)
  |               |
  |-------------->| CREATE_CHILD_SA request (HDR, SK {SA, Ni, TSi, TSr})
  |               | (Judgment #3)
  |               |
  V               V
```

Packet #1	See below
Packet #2	See Common Packet #6

Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #6
UDP Header	Same as the Common Packet #6
IKEv2 Header	Same as the Common Packet #6

E Payload	Same as the Common Packet #6	
IDi Payload	Same as the Common Packet #6	
AUTH Payload	Same as the Common Packet #6	
N Payload	Same as the Common Packet #6	
SA Payload	Same as the Common Packet #6	
TSi Payload	Other fields are same as the Common Packet #6	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #6	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

Part A: (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. NUT starts to negotiate new CHILD_SA with TN1 by sending CREATE_CHILD_SA request.
7. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.I.1.2.5.2: Receipt of cryptographically valid message on the new SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to g enerate new CHILD_SAs.

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3
- [RFC 4718] - Sections 4.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1            NUT            TN1            TH2            TH3
(Host)         (SGW)          (SGW)          (Host)         (Host)
  |              |              |              |              |
  |              |------------->|              |              | IKE_SA_INIT request (HDR, SAi1,
  |              |              |              |              |                          KEi, Ni)
  |              |              |              |              | (Judgment #1)
  |              |<-------------|              |              | IKE_SA_INIT Response (HDR, SAr1,
  |              |              |              |              |                           KEr, Nr)
  |              |              |              |              | (Packet #1)
  |              |              |              |              |
  |              |------------->|              |              | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |              |              |              |              |                  SAi2, TSi, TSr})
  |              |              |              |              | (Judgment #2)
  |              |<-------------|              |              | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |              |              |              |              |                   SAr2, TSi, TSr})
  |              |              |              |              | (Packet #2)
  |              |              |              |              |
  |<-------------+==============+--------------|              | IPsec {Echo Request}
  |              |              |              |              | (Packet #3) (Judgment #3)
  |--------------+==============+------------->|              | IPsec {Echo Reply}
  |              |              |              |              | (Packet #4) (Judgment #4)
  |              |              |              |              |
  |       X------+==============+-----------------------------| IPsec {Echo Request}
  |              |              |              |              | (Packet #5) (Judgment #5)
  |--------------+=======X      |              |              | IPsec {Echo Request}
  |              |              |              |              | (Packet #6) (Judgment #6)
  |              |              |              |              |
  |              |------------->|              |              | CREATE_CHILD_SA request (HDR,
  |              |              |              |              |                SK{SA, Ni, TSi, TSr})
  |              |              |              |              | (Judgment #7)
  |              |<-------------|              |              | CREATE_CHILD_SA response (HDR,
  |              |              |              |              |                SK{SA, Nr, TSi, TSr})
  |              |              |              |              | (Packet #7)
  |              |              |              |              |
  |<-------------+==============+--------------|              | IPsec {Echo Request}
  |              |              |              |              | (Packet #8) (Judgment #8)
  |--------------+==============+------------->|              | IPsec {Echo Reply}
  |              |              |              |              | (Packet #9) (Judgment #9)
  |              |              |              |              |
  |<-------------+==============+-----------------------------| IPsec {Echo Request}
  |              |              |              |              | (Packet #10) (Judgment #10)
  |--------------+==============+---------------------------->| IPsec {Echo Reply}
  |              |              |              |              | (Packet #11) (Judgment #11)
  |              |              |              |              |
  V              V              V              V              V
```

Packet #1	See Common Packet #2
Packet #2	See below
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 5.
Packet #6	See below
Packet #7	See below
Packet #8	See Common Packet #21
Packet #9	See Common Packet #25
Packet #10	See below This packet is cryptographically protected by the

	CHILD_SA negotiated at Step 14 to Step 16.
Packet #11	See below

- Packet #2: IKE_AUTH response

IPv6 Header	Same as the Common Packet #4	
UDP Header	Same as the Common Packet #4	
IKEv2 Header	Same as the Common Packet #4	
E Payload	Same as the Common Packet #4	
IDi Payload	Same as the Common Packet #4	
AUTH Payload	Same as the Common Packet #4	
N Payload	Same as the Common Packet #4	
SA Payload	Same as the Common Packet #4	
TSi Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH1's Global Address on Link B
		Ending Address	TH1's Global Address on Link B

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH2's Global Address on Link Y
		Ending Address	TH2's Global Address on Link Y

- Packet #5: Echo Request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #6: Echo Request

IPv6 Header	Source Address	TH1's Global Address
	Distination Address	TH3's Global Address
ICMPv6 Header	Type	128
	Code	0

	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #7: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #4	
UDP Header	Same as the Common Packet #4	
IKEv2 Header	Same as the Common Packet #4	
E Payload	Same as the Common Packet #4	
IDi Payload	Same as the Common Packet #4	
AUTH Payload	Same as the Common Packet #4	
N Payload	Same as the Common Packet #4	
SA Payload	Same as the Common Packet #4	
TSi Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH1’s Global Address on Link B
		Ending Address	TH1’s Global Address on Link B

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH3’s Global Address on Link Y
		Ending Address	TH3’s Global Address on Link Y

- Packet #10: Echo Request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT’s Global Address on Link A
ESP	Security Parameter Index	CHILD_SA’s SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet’s Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #11: Echo Reply

IPv6 Header	Source Address	TH1's Global Address
	Distination Address	TH3's Global Address
ICMPv6 Header	Type	129
	Code	0

	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TH2 transmits an Echo Request packet to TH1.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply packet to TH2.
9. Observe the messages transmitted on Link B.
10. TH3 transmits an Echo Request packet to TH1.
11. Observe the messages transmitted on Link A.
12. TH1 transmits an Echo Request packet to TH3.
13. Observe the messages transmitted on Link B.
14. NUT starts to negotiate new CHILD_SA with TN1 by sending CREATE_CHILD_SA request.
15. Observe the messages transmitted on Link B.
16. After a reception of CREATE_CHILD_SA request from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT with following Traffic Selector
17. TH2 transmits an Echo Request packet to TH1.
18. Observe the messages transmitted on Link A.
19. TH1 transmits an Echo Reply packet to TH2.
20. Observe the messages transmitted on Link B.
21. TH3 transmits an Echo Request packet to TH1.
22. Observe the messages transmitted on Link A.
23. TH1 transmits an Echo Reply packet to TH3.
24. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Request with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT never forwards an Echo Request.

Step 13: Judgment #6
The NUT never forwards an Echo Request.

Step 15: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT forwards an Echo Request.

Step 13: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 2.6. Exchange Collisions

Test IKEv2.SGW.I.1.2.6.1: Simultaneous CHILD_SA Close

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA message to close CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<------------------|                   | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<------------------|                   | IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
  |                   |                   *                   | wait until CHILD_SA expires
  |                   |                   |                   |
  |                   |------------------>|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Judgment #3)
  |                   |                   |                   |
  |                   |<------------------|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Packet #3)
  |                   |                   |                   |
  |                   |<------------------|                   | INFORMATIONAL response (HDR, SK { })
  |                   |                   |                   | (Packet #4)
  |                   |                   |                   |
  |                   |------------------>|                   | INFORMATIONAL response (HDR, SK { })
  |                   |                   |                   | (Judgment #4)
  |                   |                   |                   |
  |                   |         X=======+-------------------| IPsec {Echo Request}
  |                   |                   |                   | (Packet #5) (Judgment #5)
  |-------------------+=======X           |                   | IPsec {Echo Reply}
  |                   |                   |                   | (Packet #6) (Judgment #6)
  |                   |                   |                   |
  V                   V                   V                   V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See below
Packet #4	See Common Packet #19
Packet #5	See Common Packet #21
Packet #6	See Common Packet #25

Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)

	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request to close CHILD_SA established at Step 5.
9. TN1 responds with an INFROMATIONAL response with no payload to an INFORMATIONAL resquest received at Step 7.
10. Observe the messages transmitted on Link A.
11. TH2 transmits an Echo Request to TH1.
12. Observe the messages transmitted on Link B.
13. TH1 transmits an Echo Reply to TH2.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Step 10: Judgment #4
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close CHILD_SA.

Step 12: Judgment #5
The NUT forwards an Echo Request.

Step 14: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.2: Simultaneous IKE_SA Close

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA message to close IKE_SA.

References:

- [RFC 4718] - Sections 5.11.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Judgment #1)
   |<--------------| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Packet #1)
   |               |
   |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Judgment #2)
   |<--------------| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Packet #2)
   |               |
   |               * wait until expiring CHILD_SA
   |               |
   |-------------->| INFORMATIONAL request (HDR, SK {D})
   |               | (Judgment #3)
   |               |
   |<--------------| INFORMATIONAL request (HDR, SK {D})
   |               | (Packet #3)
   |               |
   |<--------------| INFORMATIONAL response (HDR, SK { })
   |               | (Packet #4)
   |               |
   |-------------->| INFORMATIONAL response (HDR, SK { })
   |               | (Judgment #4)
   |               |
   |<--------------| INFORMATIONAL request (HDR, SK { })
   |               | (Packet #5)
   |-------X       | INFORMATIONAL response (HDR, SK { })
   |               | (Judgment #5)
   V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See below
Packet #4	See Common Packet #17
Packet #5	See Common Packet #17

Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 waits until expiring IKE_SA's lifetime and does not respond to an INFORMATIONAL request with an INFORMATIONAL response for liveness check.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request to close CHILD_SA established at Step 5.
9. TN1 responds with an INFROMATIONAL response with no payload to an INFORMATIONAL response received at Step 7.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is

cryptographically protected by IKE_SA to be closed.

12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inbound SPI value to be deleted as SPI.

Step 10: Judgment #4
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close CHILD_SA.

Step 12: Judgment #5
The NUT never transmits an INFORMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.3: Simultaneous CHILD_SA Rekeying

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<---------------|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |--------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                  |                  |                  |                           SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<---------------|                  | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                  |                  |                  |                            SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
 ...                ...                ...                ...
  |                  |                  |                  |                         ---
  |<---------------+================+---------------| IPsec {Echo Request}      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |---------------+================+--------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                         ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |--------------->|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |<---------------|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Packet #5)
  |                  |--------------->|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                  |                  |                  | (Judgment #6)
  |                  |                  |                  |
  |                  |<---------------|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                  |                  |                  | (Packet #6)
  |                  |                  |                  |
  |                  |--------------->|                  | INFORMATIONAL request (HDR, SK {D})
  |                  |                  |                  | (Judgment #7)
  |                  |<---------------|                  | INFORMATIONAL response (HDR, SK {D})
  |                  |                  |                  | (Packet #7)
  |                  |                  |                  |
  |                  |--------------->|                  | INFORMATIONAL request (HDR, SK {D})
  |                  |                  |                  | (Judgment #8)
  |                  |<---------------|                  | INFORMATIONAL response (HDR, SK {D})
  |                  |                  |                  | (Packet #8)
  |                  |                  |                  |
  |<---------------+================+---------------| IPsec {Echo Request}  (new CHILD_SA)
  |                  |                  |                  | (Packet #9) (Judgment #9)
  |---------------+================+--------------->| IPsec {Echo Reply}    (new CHILD_SA)
  |                  |                  |                  | (Packet #10) (Judgment #10)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See Common Packet #16
Packet #7	See below

Packet #8	See below
Packet #9	See Common Packet #21
Packet #10	See Common Packet #25

Packet #7: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Packet #8: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0

	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the new CHILD_SA initiated by the NUT at Step 9

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA expires.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey CHILD_SA to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with a CREATE_CHILD_SA response to the CRETE_CHILD_SA received at Step 9. The response message includes minimum Nonce Data.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an INFORMATIONAL response to the INFORMATIONAL request received at Step 15.
17. Observe the messages transmitted on Link A.
18. TN1 responds with an INFORMATIONAL response to the INFORMATIONAL request received at Step 17.
19. TH2 transmits an Echo Request to TH1.
20. Observe the messages transmitted on Link B.
21. TH1 transmits an Echo Reply to TH2.
22. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 15: Judgment #7
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the original CHILD_SA.

Step 18: Judgment #8
The NUT transmits an INFORMATIONAL request with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the new CHILD_SA initiated by the NUT at Step 11.

Step 20: Judgment #9
The NUT forwards an Echo Request.

Step 22: Judgment #10
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.4: Simultaneous CHILD_SA Rekeying with retransmission

Purpose:

To verify an IKEv2 device properly handles simultaneous CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<------------------|                   | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                   |                   |                   |                            SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<------------------|                   | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                   |                   |                   |                             SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                          ---
  |<------------------+===================+-------------------| IPsec {Echo Request}      | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
  |-------------------+===================+------------------>| IPsec {Echo Reply}        | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4) |
  |                   |                   |                   |                          ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |------------------>|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Judgment #5)
  |                   |                   |                   |
  |                   |<------------------|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Packet #5)
  |                   |------------------>|                   | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                   |                   |                   | (Judgment #6)
  |                   |                   |                   |
  |                   |<------------------|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Packet #6)
  |                   |------------------>|                   | INFORMATIONAL response (HDR, SK {D})
  |                   |                   |                   | (Judgment #7)
  |                   |                   |                   |
  |                   |------------------>|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Judgment #8)
  |                   |<------------------|                   | CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)}
  |                   |                   |                   | (Packet #7)
  |                   |                   |                   |
  |<------------------+===================+-------------------| IPsec {Echo Request}  (new CHILD_SA)
  |                   |                   |                   | (Packet #8) (Judgment #9)
  |-------------------+===================+------------------>| IPsec {Echo Reply}    (new CHILD_SA)
  |                   |                   |                   | (Packet #9) (Judgment #10)
  |                   |                   |                   |
  V                   V                   V                   V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See below
Packet #7	See below
Packet #8	See Common Packet #21
Packet #9	See Common Packet #25

Packet #6: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Packet #7: CREATE_CHILD_SA response

IPv6 Header	Same as Common Packet #14	
UDP Header	Same as Common Packet #14	
IKEv2 Header	Same as Common Packet #14	
E Payload	Same as Common Packet #14	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	10
	Protocol ID	0
	SPI Size	0
	Notify Message Type	NO_PROPOSAL_CHOSEN (14)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.

8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey CHILD_SA to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 transmits an INFORMAITONAL request with a Delete Payload to close the replaced CHILD_SA.
15. Observe the messages transmitted on Link A.
16. Observe the messages transmitted on Link A.
17. TN1 responds with a CREATE_CHILD_SA response with a Notify payload of type NO_PROPOSAL_CHOSEN to the retransmitted CREATE_CHILD_SA request.
18. TH2 transmits an Echo Request to TH1.
19. Observe the messages transmitted on Link B.
20. TH1 transmits an Echo Reply to TH2.
21. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA's SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 15: Judgment #7
The NUT transmits an INFORMATIONAL response with a Delete Payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value of the original CHILD_SA.

Step 16: Judgment #8

The NUT retransmits the same CREATE_CHILD_SA request as the message at Step 11. The message includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 19: Judgment #9
The NUT forwards an Echo Request.

Step 21: Judgment #10
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.5: Simultaneous IKE_SA Rekeying

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4718] - Sections 5.11.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1           NUT           TN1           TH2
(Host)        (SGW)         (SGW)         (Host)
  |             |             |             |
  |             |------------>|             | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |             |             |             | (Judgment #1)
  |             |<------------|             | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |             |             |             | (Packet #1)
  |             |             |             |
  |             |------------>|             | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |             |             |             |                     SAi2, TSi, TSr})
  |             |             |             | (Judgment #2)
  |             |<------------|             | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |             |             |             |                      SAr2, TSi, TSr})
  |             |             |             | (Packet #2)
  |             |             |             |
 ...           ...           ...           ...
  |             |             |             |                           ---
  |<------------+=============+-------------| IPsec {Echo Request}     | repeat Echo exchange
  |             |             |             | (Packet #3) (Judgment #3) | until lifetime of SA
  |-------------+=============+------------>| IPsec {Echo Reply}       | is expired
  |             |             |             | (Packet #4) (Judgment #4) |
  |             |             |             |                           ---
 ...           ...           ...           ...
  |             |             |             |
  |             |------------>|             | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |             |             |             | (Judgment #5)
  |             |             |             |
  |             |<------------|             | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |             |             |             | (Packet #5)
  |             |------------>|             | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |             |             |             | (Judgment #6)
  |             |             |             |
  |             |<------------|             | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |             |             |             | (Packet #6)
  |             |             |             |
  |             |------------>|             | INFORMATIONAL request (HDR, SK {D})
  |             |             |             | (Judgment #7)
  |             |<------------|             | INFORMATIONAL response (HDR, SK {})
  |             |             |             | (Packet #7)
  |             |             |             |
  |             |------------>|             | INFORMATIONAL request (HDR, SK {D})
  |             |             |             | (Judgment #8)
  |             |<------------|             | INFORMATIONAL response (HDR, SK {})
  |             |             |             | (Packet #8)
  |             |             |             |
  |             |<------------|             | INFORMATIONAL request (HDR, SK {})  (new IKE_SA)
  |             |             |             | (Packet #9)
  |             |------------>|             | INFORMATIONAL response (HDR, SK {}) (new IKE_SA)
  |             |             |             | (Judgment #9)
  V             V             V             V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #11
Packet #6	See Common Packet #12
Packet #7	See Common Packet #18
Packet #8	See Common Packet #18
Packet #8	See Common Packet #17

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 responds with a CREATE_CHILD_SA response to the CREATE_CHILD_SA request received at Step 11. The response message includes minimum Nonce Data to make the NUT send a message to close duplicated IKE_SA.
15. Observe the messages transmitted on Link A.
16. TN1 responds with an INFORMATIONAL response with no payload.
17. Observe the messages transmitted on Link A.
18. TN1 responds with an INFORMATIONAL response with no payload.
19. TN1 transmits an INFORMATIONAL request with no payload to the NUT. The message is cryptographically protected by the new IKE_SA initiated by TN1 at Step 12.
20. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. And the proposal in the SA payload Response has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA's responder's SPI value in the SPI field.

Step 15: Judgment #7
The NUT transmits an INFORMATIONAL request . The message's IKE_SA Initiator's SPI value is the IKE_SA Initiator's SPI value of the original IKE_SA, and the message's IKE_SA Responder's SPI value is the IKE_SA Responder's SPI value of the original IKE_SA. The message also has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Step 17: Judgment #8
The NUT transmits an INFORMATIONAL request . The message's IKE_SA Initiator's SPI value is the IKE_SA Initiator's SPI value of the new IKE_SA initiated by the NUT at Step 9, and the message's IKE_SA Responder's SPI value is the IKE_SA Responder's SPI value of the new IKE_SA initiated by the NUT at Step 9. The message also has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.

Step 20: Judgment #9
The NUT transmits an INFOMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.6: Simultaneous IKE_SA Rekeying with retransmission

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4718] - Sections 5.11.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 60 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1          NUT          TN1          TH2
(Host)       (SGW)        (SGW)        (Host)
  |            |            |            |
  |            |----------->|            | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |            |            |            | (Judgment #1)
  |            |<-----------|            | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |            |            |            | (Packet #1)
  |            |            |            |
  |            |----------->|            | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |            |            |            |                         SAi2, TSi, TSr})
  |            |            |            | (Judgment #2)
  |            |<-----------|            | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |            |            |            |                          SAr2, TSi, TSr})
  |            |            |            | (Packet #2)
  |            |            |            |
 ...          ...          ...          ...
  |            |            |            |                              ---
  |<-----------+============+------------| IPsec {Echo Request}        | repeat Echo exchange
  |            |            |            | (Packet #3) (Judgment #3)   | until lifetime of SA
  |------------+============+----------->| IPsec {Echo Reply}          | is expired
  |            |            |            | (Packet #4) (Judgment #4)   |
  |            |            |            |                              ---
 ...          ...          ...          ...
  |            |            |            |
  |            |----------->|            | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |            |            |            | (Judgment #5)
  |            |            |            |
  |            |<-----------|            | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |            |            |            | (Packet #5)
  |            |----------->|            | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |            |            |            | (Judgment #6)
  |            |            |            |
  |            |<-----------|            | INFORMATIONAL request (HDR, SK {D})
  |            |            |            | (Packet #6)
  |            |----------->|            | INFORMATIONAL response (HDR, SK {})
  |            |            |            | (Judgment #7)
  |            |            |            |
  |            |-------X    |            | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |            |            |            | (Judgment #8)
  V            V            V            V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #11
Packet #6	See below

Packet #6: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any

	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA to the NUT.
13. Observe the messages transmitted on Link A.
14. TN1 transmits an INFORMATONAL request to close the original IKE_SA. The message has a Delete Payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.
15. Observe the messages transmitted on Link A.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the proposal in the SA payload Response has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s responder’s SPI value in the SPI field.

Step 15: Judgment #7
The NUT responds with an INFOMATIONAL response to the INFORMATIONAL request to close the original IKE_SA.

Step 16: Judgment #8
The NUT never retransmits a CREATE_CHILD_SA request transmitted at Step 11.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.7: Rekeying a CHILD_SA while Closing a CHILD_SA

Purpose:

To verify an IKEv2 device properly handles simultaneous closing and rekeying a CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
    NUT              TN1
   (SGW)            (SGW)
     |                |
     |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
     |                | (Judgment #1)
     |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
     |                | (Packet #1)
     |                |
     |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
     |                | (Judgment #2)
     |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
     |                | (Packet #2)
     |                |
     |                * wait until CHILD_SA expires
     |                |
     |--------------->| INFORMATIONAL request (HDR, SK {D})
     |                | (Judgment #3)
     |                |
     |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Nr, TSi, TSr})
     |                | (Packet #3)
     |                |
     |<---------------| INFORMATIONAL response (HDR, SK {D})
     |                | (Packet #4)
     |                |
     |--------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
     |                | (Judgment #4)
     |                |
     V                V

 N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Packet #3	See Common Packet #15
Packet #4	See below

Packet #4: INFORMATIONAL response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey a CHILD_SA.
8. TN1 responds with an INFORMATIONAL response to an INFORMATIONAL request to close a CHILD_SA.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATONAL request to close a CHILD_SA.

Step 9: Judgment #4
The NUT responds with a CREATE_CHILD_SA response to a CREATE_CHILD_SA reqest to rekey a CHILD_SA. The CREATE_CHILD_SA response includes a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.8: Closing a New CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a request to close nonexistent CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.6

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1               NUT               TN1               TH2
(Host)            (SGW)             (SGW)             (Host)
  |                 |                 |                 |
  |                 |---------------->|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 |                 |                 | (Judgment #1)
  |                 |<----------------|                 | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 |                 |                 | (Packet #1)
  |                 |                 |                 |
  |                 |---------------->|                 | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                 |                 |                 |                          SAi2, TSi, TSr})
  |                 |                 |                 | (Judgment #2)
  |                 |<----------------|                 | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                 |                 |                 |                          SAr2, TSi, TSr})
  |                 |                 |                 | (Packet #2)
  |                 |                 |                 |
 ...               ...               ...               ...
  |                 |                 |                 |                            ---
  |<----------------+=================+-----------------| IPsec {Echo Request}        | repeat Echo exchange
  |                 |                 |                 | (Packet #3) (Judgment #3)   | until lifetime of SA
  |-----------------+=================+---------------->| IPsec {Echo Reply}          | is expired
  |                 |                 |                 | (Packet #4) (Judgment #4)   |
  |                 |                 |                 |                            ---
 ...               ...               ...               ...
  |                 |                 |                 |
  |                 |---------------->|                 | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                 |                 |                 | (Judgment #5)
  |                 |        X--------|                 | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                 |                 |                 | (Packet #5)
  |                 |                 |                 |
  |                 |<----------------|                 | INFORMATIONAL request (HDR, SK {D})
  |                 |                 |                 | (Packet #6)
  |                 |---------------->|                 | INFORMATIONAL response (HDR, SK {})
  |                 |                 |                 | (Judgment #6)
  V                 V                 V                 V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16
Packet #6	See below

Packet #6: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	0
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0

	Message ID	The same value as corresponding request's Message ID
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value of the original CHILD_SA

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA response to rekey a CHILD_SA to the NUT. But the response does not reach the NUT.
13. TN1 transmits an INFORMATIONAL request to close a CHILD_SA which were supposed to be created by rekey.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3

The NUT forwards an Echo Request.

Step 9: Judgment #4

The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 14: Judgment #6
The NUT responds with an INFORMATIONAL response with no payload to the TN1.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.9: Rekeying a New CHILD_SA

Purpose:

To verify an IKEv2 device properly handles a request to rekey nonexistent CHILD_SA.

References:

- [RFC 4718] - Sections 5.11.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT             TN1             TH2
(Host)           (SGW)           (SGW)           (Host)
  |                |               |               |
  |                |-------------->|               | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |               |               | (Judgment #1)
  |                |<--------------|               | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |               |               | (Packet #1)
  |                |               |               |
  |                |-------------->|               | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |               |               |                          SAi2, TSi, TSr})
  |                |               |               | (Judgment #2)
  |                |<--------------|               | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                |               |               |                           SAr2, TSi, TSr})
  |                |               |               | (Packet #2)
  |                |               |               |
 ...              ...             ...             ...
  |                |               |               |                          ---
  |<---------------+===============+---------------| IPsec {Echo Request}     | repeat Echo exchange
  |                |               |               | (Packet #3) (Judgment #3) | until lifetime of SA
  |----------------+===============+-------------->| IPsec {Echo Reply}       | is expired
  |                |               |               | (Packet #4) (Judgment #4) |
  |                |               |               |                          ---
 ...              ...             ...             ...
  |                |               |               |
  |                |-------------->|               | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |               |               | (Judgment #5)
  |                |       X-------|               | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                |               |               | (Packet #5)
  |                |               |               |
  |                |<--------------|               | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                |               |               | (Packet #6)
  |                |-------------->|               | CREATE_CHILD_SA response (HDR, SK
{N(NO_PROPOSAL_CHOSEN)})
  |                |               |               | (Judgment #6)
  V                V               V               V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16
Packet #6	See Common Packet #15 The SPI value in the Delete payload is the same value as the SPI value in Packet #5 SA payload.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.

9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA response to rekey a CHILD_SA to the NUT. But the response does not reach the NUT.
13. TN1 transmits a CREATE_CHILD_SA request to rekey the CHILD_SA which were supposed to be created by rekey.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 14: Judgment #6
The NUT responds with a CREATE_CHILD_SA response with a Notify of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.10: Rekeying an IKE_SA with half-open CHILD_SAs

Purpose:

To verify an IKEv2 device properly handles a request to rekey an IKE_SA which has CHILD_SAs in half-open state.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  TH1                 NUT                 TN1                 TH2
 (Host)              (SGW)               (SGW)               (Host)
   |                   |                   |                   |
   |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                   |                   |                   | (Judgment #1)
   |                   |<------------------|                   | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                   |                   |                   | (Packet #1)
   |                   |                   |                   |
   |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH,
   |                   |                   |                   |                       SAi2, TSi, TSr})
   |                   |                   |                   | (Judgment #2)
   |                   |<------------------|                   | IKE_AUTH response (HDR, SK {IDr, AUTH,
   |                   |                   |                   |                        SAr2, TSi, TSr})
   |                   |                   |                   | (Packet #2)
   |                   |                   |                   |
  ...                 ...                 ...                 ...
   |                   |                   |                   |                          ---
   |<------------------+===================+-------------------| IPsec {Echo Request}      | repeat Echo exchange
   |                   |                   |                   | (Packet #3) (Judgment #3) | until lifetime of SA
   |-------------------+===================+------------------>| IPsec {Echo Reply}        | is expired
   |                   |                   |                   | (Packet #4) (Judgment #4) |
   |                   |                   |                   |                          ---
  ...                 ...                 ...                 ...
   |                   |                   |                   |
   |                   |------------------>|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                   |                   |                   | (Judgment #5)
   |                   |                   |                   |
   |                   |<------------------|                   | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
   |                   |                   |                   | (Packet #5)
   |                   |------------------>|                   | CREATE_CHILD_SA response (HDR, SK{N(NO_PROPOSAL_CHOSEN)})
   |                   |                   |                   | (Judgment #6)
   V                   V                   V                   V

 N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA to the NUT.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey a CHILD_SA. The message includes “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.11: Rekeying a CHILD_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to rekey a CHILD_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |-------------->|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Judgment #1)
  |                  |<--------------|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Packet #1)
  |                  |                  |                  |
  |                  |-------------->|                  | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                  |                  |                  |                           SAi2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |<--------------|                  | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |                  |                  |                  |                            SAr2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |                  |                  |
...                ...                ...                ...
  |                  |                  |                  |                          ---
  |<-------------+===============+--------------| IPsec {Echo Request}      | repeat Echo exchange
  |                  |                  |                  | (Packet #3) (Judgment #3) | until lifetime of SA
  |--------------+===============+------------->| IPsec {Echo Reply}        | is expired
  |                  |                  |                  | (Packet #4) (Judgment #4) |
  |                  |                  |                  |                          ---
 ...                ...                ...                ...
  |                  |                  |                  |
  |                  |-------------->|                  | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |<--------------|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr })
  |                  |                  |                  | (Packet #5)
  |                  |-------------->|                  | CREATE_CHILD_SA response (HDR, SK{N(NO_ADDITIONAL_SAS)})
  |                  |                  |                  | (Judgment #6)
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits a CREATE_CHILD_SA request to rekey a CHILD_SA to the NUT.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_ADDTIONAL_SAS to a CREATE_CHILD_SA request to rekey a CHILD_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.12: Rekeying an IKE_SA with half-closed CHILD_SAs

Purpose:

To verify an IKEv2 device properly handles a request to rekey an IKE_SA which has CHILD_SAs in half-closed state.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Judgment #1)
   |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |               | (Packet #1)
   |               |
   |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Judgment #2)
   |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Packet #2)
   |               |
   |               * wait until CHILD_SA expires
   |               |
   |-------------->| INFORMATIONAL request (HDR, SK {D})
   |               | (Judgment #3)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
   |               | (Packet #3)
   |-------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
   |               | (Judgment #4)
   |               |
   V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.

2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA to the NUT.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATIONAL request to close a CHILD_SA to the TN1.

Step 8: Judgment #4
The NUT responds with a CREATE_CHILD_SA response which has a Notify of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.13: Closing a CHILD_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to close a CHILD_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  TH1               NUT              TN1              TH2
 (Host)            (SGW)            (SGW)            (Host)
   |                 |                |                |
   |                 |-------------->|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 |                |                | (Judgment #1)
   |                 |<--------------|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                 |                |                | (Packet #1)
   |                 |                |                |
   |                 |-------------->|                | IKE_AUTH request (HDR, SK {IDi, AUTH,
   |                 |                |                |                          SAi2, TSi, TSr})
   |                 |                |                | (Judgment #2)
   |                 |<--------------|                | IKE_AUTH response (HDR, SK {IDr, AUTH,
   |                 |                |                |                          SAr2, TSi, TSr})
   |                 |                |                | (Packet #2)
   |                 |                |                |
  ...               ...              ...              ...
   |                 |                |                |                          ---
   |<----------------+================+----------------| IPsec {Echo Request}     | repeat Echo exchange
   |                 |                |                | (Packet #3) (Judgment #3) | until lifetime of SA
   |-----------------+================+--------------->| IPsec {Echo Reply}       | is expired
   |                 |                |                | (Packet #4) (Judgment #4) |
   |                 |                |                |                          ---
  ...               ...              ...              ...
   |                 |                |                |
   |                 |-------------->|                | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
   |                 |                |                | (Judgment #5)
   |                 |                |                |
   |                 |<--------------|                | INFORMATIONAL request (HDR, SK {D})
   |                 |                |                | (Packet #5)
   |                 |-------------->|                | INFORMATIONAL response (HDR, SK{})
   |                 |                |                | (Judgment #6)
   V                 V                V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below

Packet #5: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an INFORMATIONAL request to close a CHILD_SA to the NUT.
13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT responds with an INFORMATIONAL response with no payload.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.14: Closing an IKE_SA while rekeying an IKE_SA

Purpose:

To verify an IKEv2 device properly handles a request to close an IKE_SA after IKE_SA rekey has been started.

References:

- [RFC 4718] - Sections 5.11.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1           NUT           TN1           TH2
(Host)        (SGW)         (SGW)         (Host)
  |             |             |             |
  |             |------------>|             | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |             |             |             | (Judgment #1)
  |             |<------------|             | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |             |             |             | (Packet #1)
  |             |             |             |
  |             |------------>|             | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |             |             |             |                          SAi2, TSi, TSr})
  |             |             |             | (Judgment #2)
  |             |<------------|             | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |             |             |             |                           SAr2, TSi, TSr})
  |             |             |             | (Packet #2)
  |             |             |             |
 ...           ...           ...           ...
  |             |             |             |                         ---
  |<------------+=============+-------------| IPsec {Echo Request}      | repeat Echo exchange
  |             |             |             | (Packet #3) (Judgment #3) | until lifetime of SA
  |-------------+=============+------------>| IPsec {Echo Reply}        | is expired
  |             |             |             | (Packet #4) (Judgment #4) |
  |             |             |             |                         ---
 ...           ...           ...           ...
  |             |             |             |
  |             |------------>|             | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |             |             |             | (Judgment #5)
  |             |             |             |
  |             |<------------|             | INFORMATIONAL request (HDR, SK {D})
  |             |             |             | (Packet #5)
  |             |             |             |
  |             |<------------|             | CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
  |             |             |             | (Packet #6)
  |             |             |             |
  |             |------------>|             | INFORMATIONAL response (HDR, SK{})
  |             |             |             | (Judgment #6)
  V             V             V             V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below

Packet #5: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0

	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Packet #6: CREATE_CHILD_SA response

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	36 (CREATE_CHILD_SA)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	1
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	41 (N)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	0
	SPI Size	0
	Notify Message Type	14 (NO_PROPOSAL_CHOSEN)

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.

5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 and 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. TN1 transmits an INFORMATIONAL request to close an IKE_SA to the NUT.
13. TN1 responds with a CREATE_CHILD_SA response which has a Notify payload of type NO_PROPOSAL_CHOSEN to a CREATE_CHILD_SA request to rekey an IKE_SA.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request to rekey an IKE_SA. The message includes “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the CREATE_CHILD_SA request has a SA payload including 1 (IKE) in the Protocol ID field, 8 in the SPI size field and new IKE_SA’s SPI value in the SPI field.

Step 14: Judgment #6
The NUT responds with an INFORMATIONAL response with no payload to an INFORMATIONAL request to close an IKE_SA.

Possible Problems:

- Each NUT has the different lifetime of SA.

Test IKEv2.SGW.I.1.2.6.15: Rekeying an IKE _SA while Closing an IKE_SA

Purpose:

To verify an IKEv2 device properly handles simultaneous closing and rekeying an IKE_SA.

References:

- [RFC 4718] - Sections 5.11.10

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 30 seconds and set CHILD_SA Lifetime to 300 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |-------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Judgment #1)
   |<--------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |               | (Packet #1)
   |               |
   |-------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Judgment #2)
   |<--------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Packet #2)
   |               |
   |               * wait until CHILD_SA expires
   |               |
   |-------------->| INFORMATIONAL request (HDR, SK {D})
   |               | (Judgment #3)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK { SA, Ni })
   |               | (Packet #3)
   |               |
   |-------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
   |               | (Judgment #4)
   |               |
   V               V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #11

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. Observe the messages transmitted on Link A.
7. TN1 transmits a CREATE_CHILD_SA request to rekey an IKE_SA.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT transmits an INFORMATONAL request to close an IKE_SA.

Step 8: Judgment #4
The NUT responds with a CREATE_CHILD_SA response to a CREATE_CHILD_SA reqest to rekey an IKE_SA. The CREATE_CHILD_SA response includes a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 2.7. Non zero RESERVED fields

Test IKEv2.SGW.I.1.2.7.1: Non zero RESERVED fields in CREATE_CHILD_SA response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                 NUT                 TN1                 TH2
(Host)              (SGW)               (SGW)               (Host)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   |                   |                   | (Judgment #1)
  |                   |<------------------|                   | IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                   |                   |                   | (Packet #1)
  |                   |                   |                   |
  |                   |------------------>|                   | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   |                   |                   | (Judgment #2)
  |                   |<------------------|                   | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   |                   |                   | (Packet #2)
  |                   |                   |                   |
 ...                 ...                 ...                 ...
  |                   |                   |                   |                                ---
  |<------------------+===================+-------------------| IPsec {Echo Request}         | repeat Echo exchange
  |                   |                   |                   | (Packet #3) (Judgment #3)    | until lifetime of SA
  |-------------------+===================+------------------>| IPsec {Echo Reply}           | is expired
  |                   |                   |                   | (Packet #4) (Judgment #4)    |
  |                   |                   |                   |                                ---
 ...                 ...                 ...                 ...
  |                   |                   |                   |
  |                   |------------------>|                   | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                   |                   |                   | (Judgment #5)
  |                   |                   |                   |
  |                   |<------------------|                   | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                   |                   |                   | (Packet #5)
  |                   |                   |                   |
  |                   |------------------>|                   | INFORMATIONAL request (HDR, SK {D})
  |                   |                   |                   | (Judgment #6)
  V                   V                   V                   V

N: REKEY_SA
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #16 All RESERVED fields are set to one.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP using the first negotiated algorithms to NUT.
7. Observe the messages transmitted on Link B.
8. TH1 transmits an Echo Reply to TH2.
9. Observe the messages transmitted on Link A.
10. Repeat Steps 6 through 9 until lifetime of SA is expired.
11. Observe the messages transmitted on Link A.
12. After reception of CREATE_CHILD_SA request for rekeying from the NUT, TN1 responds with a CREATE_CHILD_SA response to the NUT. All RESERVED fields in the message are set to one.

13. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotitated algorithms.

Step 11: Judgment #5
The NUT transmits a CREATE_CHILD_SA request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. And the CREATE_CHILD_SA request includes a Notify payload of type REKEY_SA containing rekeyed CHILD_SA’s SPI value in the SPI field.

Step 13: Judgment #6
The NUT transmits an INFORMATIONAL request with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the inblund SPI value to be deleted as SPI.

Possible Problems:

- Each NUT has the different lifetime of SA.

Group 3. The INFORMATIONAL Exchange

Group 3.1. Header and Payload Formats

Test IKEv2.SGW.I.1.3.1.1: Sending INFORMATIONAL Exchange

Purpose:

To verify an IKEv2 device checks whether the other endpoint is alive.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |                * wait until receiving a liveness check
  |                |
  |--------------->| INFORMATIONAL request (HDR, SK {})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH

response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link A.

Part B: Encrypted Payload Format (BASIC)

8. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
9. Observe the messages transmitted on Link A.
10. TN1 responds with an IKE_SA_INIT response to the NUT.
11. Observe the messages transmitted on Link A.
12. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
13. TN1 waits for receiving an INFORMATIONAL request with no payloads.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 131 Header format

- An IKE_SA Initiator’s SPI field is set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field is set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.
- A Next Payload field is set to Encrypted Payload (46).
- A Major Version field is set to 2.
- A Minor Version field is set to zero.

- An Exchange Type field is set to INFORMATIONAL (37).
- A Flags field is set to (00010000)2 = (16)10.
- A Message ID field is set to the value incremented the previous IKE message's Message ID by one.
- A Length field is set to the length of the message (header + payloads) in octets.

Part B

Step 9: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 1: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 14: Judgment #3
The NUT transmits an INFORMATIONAL request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 132 Encrypted payload

- A Next Payload field is set to zero.
- A Critical field is set to zero.
- A RESERVED field is set to zero.
- A Payload Length field is set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field is set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field is set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field is set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field is set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Possible Problems:

- None

Group 3.2. Use of Retransmission Timers

Test IKEv2.SGW.I.1.3.2.1: Retransmission of INFORMATIONAL request

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
- In each part, configure the devices according to the Common Configuration. Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                 |
  |---------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Judgment #1)
  |<----------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                 | (Packet #1)
  |                 |
  |---------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Judgment #2)
  |<----------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 | (Packet #2)
  |                 |
  |                 * wait until receiving a liveness check
  |                 |
  |---------------->| INFORMATIONAL request (HDR, SK {})
  |                 | (Judgment #3)
  |                 |
  |                 * wait for the event of a timeout
  |                 |
  |---------------->| INFORMATIONAL request (HDR, SK {})
  |                 | (Judgment #4)
  |                 |
  V                 V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 waits for reception of IKE_AUTH response from the NUT.
6. TN1 waits for reception of INFORMATIONAL request for liveness check from the NUT.
7. Observe the messages transmitted on Link A.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it. And the request has the same Message ID value as the request received at Step 7.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Test IKEv2.SGW.I.1.3.2.2: Stop of retransmission of INFORMATIONAL request

Purpose:

To verify an IKEv2 device stops retransmission when it receives the corresponding response.

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration. Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Packet #2)
  |                |
  |                * wait until receiving a liveness check
  |                |
  |--------------->| INFORMATIONAL request (HDR, SK {})
  |                | (Judgment #3)
  |                |
  |                * wait for the event of a timeout
  |                |
  |--------------->| INFORMATIONAL request (HDR, SK {})
  |                | (Judgment #4)
  |<---------------| INFORMATIONAL response (HDR, SK {})
  |                | (Packet #3)
  |                |
  |                * wait for the event of a timeout
  |                |
  |-------X        | never send INFORMATIONAL request (HDR, SK {})
  |                | (Judgment #5)
  |                |
  V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link B.
5. TN1 waits for reception of IKE_AUTH response from the NUT.
6. TH2 transmits an Echo Request to TH1, then TN1 forwards an Echo Request with invalid SPI.
7. Observe the messages transmitted on Link B.
8. TN1 waits for the event of a timeout on NUT.
9. Observe the messages transmitted on Link B.
10. After reception of an INFORMATIONAL request from the NUT, TN1 responds with an INFORMATIONAL response to the NUT.
11. TN1 waits for the event of a timeout on NUT.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT transmits an INFORMATIONAL request followed by an Encrypted payload with no payloads contained in it. And the request has the same Message ID value as the request received at Step 7.

Step 12: Judgment #5
The NUT never retransmits an INFORMATIONAL request which has the same Message ID value as the received Step 9.

Possible Problems:

- Each NUT has the different retransmission timers. If it is imposibble to configure the retransmission timer, modifying tester is required.

Group 3.3. Non zero RESERVED fields

Test IKEv2.SGW.I.1.3.3.1: Non zero RESERVED fields in INFORMATIONAL response

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  In addition, set IKE_SA Lifetime and CHILD_SA Lifetime to more than twice as INFORMATIONAL message retransmission timer as.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Judgment #1)
   |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
   |                | (Packet #1)
   |                |
   |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                | (Judgment #2)
   |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                | (Packet #2)
   |                |
   |                * wait until receiving a liveness check
   |                |
   |--------------->| INFORMATIONAL request (HDR, SK {})
   |                | (Judgment #3)
   |<---------------| INFORMATIONAL response (HDR, SK {})
   |                | (Packet #3)
   |                |
   |-------X        | INFORMATIONAL request (HDR, SK {})
   |                | (Judgment #3)
   |                |
   V                V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #6
Packet #3	See Common Packet #18

	All RESERVED fields are set to one.

Part A: (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT.
6. TN1 waits for receiving an INFORMATIONAL request with no payloads.
7. Observe the messages transmitted on Link A.
8. TN1 responds with an INFORMATIONAL response with no payload to the NUT. All RESERVED fields in the message are set to one.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFOMATIONAL request followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #4
The NUT never retransmit an INFORMATIONAL request.

Possible Problems:

- None

Group 3.4. Error Handling

Test IKEv2.SGW.I.1.3.4.1: INVALID_SPI

Purpose:

To verify an IKEv2 device properly handles ESP packet with invalid SPI.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1          NUT           TN1          TH2
(Host)       (SGW)         (SGW)        (Host)
  |            |             |            |
  |            |------------>|            | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |            |             |            | (Judgment #1)
  |            |<------------|            | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |            |             |            | (Packet #1)
  |            |             |            |
  |            |------------>|            | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |            |             |            |                       SAi2, TSi, TSr})
  |            |             |            | (Judgment #2)
  |            |<------------|            | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |            |             |            |                        SAr2, TSi, TSr})
  |            |             |            | (Packet #2)
  |            |             |            |
  |      X-----+=============+------------| IPsec {Echo Request}
  |            |             |            | (Packet #3) (Judgment #3)
  |            |------------>|            | INFORMATIONAL request (HDR, SK {N(INVALID_SPI)})
  |            |             |            | (Judgment #4)
  |            |             |            |
  V            V             V            V
```

Packet #1	See Common Packet #2
Packet #2	See Common Packet #4
Packet #3	See Common Packet #21 This packet has an invalid SPI value (the properly netotiated value plus 1).

Part A (ADVANCED)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms. The message's SPI is set to the value of the SPI negotiated in the initial exchange plus 1.
7. Observe the messages transmitted on Link A.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT never forwards an Echo Request.

Step 8: Judgment #3
The NUT transmits an INFORMATIONAL request with a Notify payload of type INVALID_SPI. The Notify payload includes the SPI value which is transmitted at Step 6.

Possible Problems:

- None.

Section 2.1.2. Endpoint to Security Gateway Tunnel

Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.SGW.I.2.1.1.1: Sending IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)         (End-Node)
  |                |
  |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Judgment #1)
  |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                | (Packet #1)
  |                |
  |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packets

Part A: IKE Header Format (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.

Part B: Encrypted Payload Format (BASIC)

5. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link B.
7. TN1 responds with an IKE_SA_INIT response to the NUT.

8. Observe the messages transmitted on Link B.

Part C: IDi Payload Format (BASIC)

9. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link B.
11. TN1 responds with an IKE_SA_INIT response to the NUT.
12. Observe the messages transmitted on Link B.

Part D: AUTH Payload Format (BASIC)

13. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link B.
15. TN1 responds with an IKE_SA_INIT response to the NUT.
16. Observe the messages transmitted on Link B.

Part E: SA Payload Format (BASIC)

17. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link B.
19. TN1 responds with an IKE_SA_INIT response to the NUT.
20. Observe the messages transmitted on Link B.

Part F: TSi Payload Format (BASIC)

21. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link B.
23. TN1 responds with an IKE_SA_INIT response to the NUT.
24. Observe the messages transmitted on Link B.

Part G: TSr Payload Format (BASIC)

25. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link B.
27. TN1 responds with an IKE_SA_INIT response to the NUT.
28. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 133 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_AUTH (35).
- A Flags field set to (00010000)2 = (1610).
- A Message ID field set to 1.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 134 Encrypted payload

- A Next Payload field set to IDi Payload (35).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 135 ID Payload format

- A Next Payload field set to AUTH Payload (39).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field set to ID_IPV6_ADDR (5).
- A RESERVED field set to zero.
- An Identification Data field set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 136 AUTH Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field set to Shared Key Message Integrity Code (2).
- A RESERVED field set to zero.
- An Authentication Data field set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2

```
                                1                   2                   3
            0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
           ! Next    44   !0!      0      ! Length          40            !               |
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---           |
           !      0       !        0      ! Length          36            !  |            |
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
           ! Number    1  ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3  !  |            |
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
           ! SPI value                                                    !  |            |
       --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
          |!      3       !        0      ! Length          8             !  |            |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |SA Payload
          |! Type  1  (EN) !       0      ! Transform ID     3     (3DES) !  | Proposal   |
       --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
          |!      3       !        0      ! Length          8             !  |            |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
          |! Type  3  (IN) !       0      ! Transform ID     2     (SHA1) !  |            |
       --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
          |!      0       !        0      ! Length          8             !  |            |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |            |
          |! Type  5  (ESN)!       0      ! Transform ID     0       (No) !  |            |
       --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 137 SA Payload contents

The NUT transmits an IKE_AUTH request including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 138 SA Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

The following proposal must be included in Proposals field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 139 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero if this structure is the last proposal, otherwise set to 2.
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1 if this structure is the first proposal, otherwise set to 1 greater thatn the previous proposal.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 140 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.

- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 24: Judgment #2

The NUT transmits an IKE_AUTH request including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 141 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 142 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- A Ending Address field set to greater thatn or equal to Prefix B.

Part G

Step 26: Judgment #1
The NUT transmits an IKE_SA_INIT request including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2
The NUT transmits an IKE_AUTH request including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 143 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

The following traffic selector must be included in Traffic Selectors field.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 144 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to TN1 address.
- An Ending Address field set to less than or equal to TN1 address.

Possible Problems:

- IKE_AUTH request has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDi,
[CERT+],
[N(INITIAL_CONTACT)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
[IDr],
AUTH,
[CP(CFG_REQUEST)],
[N(IPCOMP_SUPPORTED)+],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA,
TSi,
TSr,
[V+]
```

- The implementation may not set single proposal by the implementation policy. In this case, Security Association Payload contains multiple proposals.

- The implementation may not set single traffic selector by the implementation policy. In this case, Traffic Selector Payload contains multiple proposals.

- Each of transforms can be located in the any order.

Test IKEv2.SGW.I.2.1.1.2: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1
(Host)           (SGW)          (End-Node)
  |                |                |
  |                |--------------->| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                | (Judgment #1)
  |                |<---------------| IKE_SA_INIT Response (HDR, SAr1, KEr, Nr)
  |                |                | (Packet #1)
  |                |                |
  |                |--------------->| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                | (Judgment #2)
  |                |<---------------| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |                | (Packet #2)
  |                |                |
  |<---------------+================| IPsec {Echo Request}
  |                |                | (Judgment #3)
  |----------------+===============>| IPsec {Echo Reply}
  |                |                | (Judgment #4)
  |                |                |
  V                V                V
```

Packet #1	See Common Packets
Packet #2	See Common Packets

Part A (BASIC)

1. NUT starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. TN1 responds with an IKE_SA_INIT response to the NUT.
4. Observe the messages transmitted on Link B.
5. After reception of IKE_AUTH request from the NUT, TN1 responds with an IKE_AUTH response to the NUT
6. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to NUT.
7. Observe the messages transmitted on Link A.
8. TH1 transmits an Echo Reply to TN1.
9. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT request including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH request including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 7: Judgment #3
The NUT forwards an Echo Request.

Step 9: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TN1 can send Echo Reply to TH1 instead of sending Echo Request.

Section 2.2. Responder
Section 2.2.1. Security Gateway to Security Gateway Tunnel
Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.SGW.R.1.1.1.1: Sending IKE_SA_INIT response

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT response using properly Header and Payloads format

References:

- [RFC4306] - Section 1.2, 2.10, 3.1, 3.2, 3.3, 3.4 and 3.9
- [RFC 4718] - Sections 7.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                | (Judgment #1)
   V                V
```

Packet #1	See Common Packet #1

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..

Part B: SA Payload Format (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
4. Observe the messages transmitted on Link A..

Part C: KE Payload Format (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..

Part D: Nonce Payload Format (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 145 Header format

- An IKE_SA Initiator’s SPI field set to IKE_SA Initiator’s SPI field value supplied in the first IKE_SA_INIT request message.
- An IKE_SA Responder’s SPI field set to a 64-bits value chosen by the NUT. It MUST not be zero.
- A Next Payload field set to SA Payload (33).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_SA_INIT (34).
- A Flags field set to (00000100)2 = (4)10.
- A Message ID field set to zero.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 4: Judgment #1

```
                               1                   2                   3
           0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
          ! Next    34    !0!      0      ! Length          44            !             |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
          !       0       !       0       ! Length          40            ! |           |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
          ! Number    1   ! Prot ID    1  ! SPI Size  0   ! Trans Cnt  4  ! |           |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |!       3       !       0       ! Length          8             ! |           |
Transform|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |! Type  1  (EN) !       0       ! Transform ID     3     (3DES) ! |           |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |!       3       !       0       ! Length          8             ! |           |SA Payload
Transform|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Proposal   |
         |! Type  2  (PR) !       0       ! Transform ID     2     (SHA1) ! |           |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |!       3       !       0       ! Length          8             ! |           |
Transform|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |! Type  3  (IN) !       0       ! Transform ID     2     (SHA1) ! |           |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |!       0       !       0       ! Length          8             ! |           |
Transform|+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |           |
         |! Type  4  (DH) !       0       ! Transform ID     2     (1024) ! |           |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ -------------
```

Figure 146 SA Payload contents

The NUT transmits an IKE_SA_INIT response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 147 SA Payload format

- A Next Payload field set to KE Payload (34).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

A Proposals field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 148 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero (last).
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 40 bytes for this proposal according to Common Configuration.
- A Proposal # field set to 1.
- A Protocol ID field set to IKE (1).
- A SPI Size field set to zero.
- A # of Transforms field set to 4.

A Transform field set to following (There are 4 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 149 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including

Header and Attribute. It is 8 bytes for PRF_HMAC_SHA1.

- A Transform Type field set to PRF (2).
- A RESERVED field set to zero.
- A Transform ID set to PRF_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #4

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for 1024 MODP Group.
- A Transform Type field set to D-H (4).
- A RESERVED field set to zero.
- A Transform ID set to Group2 (2).

Part C

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including properly formatted KE Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          DH Group #           !           RESERVED            !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Key Exchange Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 150 KE Payload format

- A Next Payload field set to Nonce Payload (40).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 136 bytes for Group 2.
- A DH Group field set to Group2 (2).
- A RESERVED field set to zero.
- A Key Exchange Data field set to Diffie-Hellman public value. The length of the Key Exchange Data field must be equal to 1024bit.

Part D

Step 8: Judgment #4

The NUT transmits an IKE_SA_INIT response including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 151 Nonce Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Nonce Data field set to random data generated by the transmitting entity. The size of the Nonce must between 16 and 256 octets.

Possible Problems:

- IKE_SA_INIT response has following packet format. It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
SA, KE, Nr,
[N(NAT_DETECTION_SOURCE_IP),
 N(NAT_DETECTION_DESTINATION_IP)],
[[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
[V+]
```

- Each of transforms can be located in the any order.

Test IKEv2.SGW.R.1.1.1.2: Sending IKE_AUTH response

Purpose:

To verify an IKEv2 device transmits IKE_AUHT response using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..

Part B: Encrypted Payload Format (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
8. Observe the messages transmitted on Link A..

Part C: IDr Payload Format (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
12. Observe the messages transmitted on Link A..

Part D: AUTH Payload Format (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A..
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A..

Part E: SA Payload Format (BASIC)

17. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A..
19. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
20. Observe the messages transmitted on Link A..

Part F: TSi Payload Format (BASIC)

21. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
22. Observe the messages transmitted on Link A..
23. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
24. Observe the messages transmitted on Link A..

Part G: TSr Payload Format (BASIC)

25. TN1starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A..
27. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
28. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 152 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_AUTH (35).
- A Flags field set to (00000100)2 = (4)10.
- A Message ID field set to 1.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 153 Encrypted payload

- A Next Payload field set to IDr Payload (36).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 154 ID Payload format

- A Next Payload field set to AUTH Payload (39).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field set to ID_IPV6_ADDR (5).
- A RESERVED field set to zero.
- An Identification Data field set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 155 AUTH Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field set to Shared Key Message Integrity Code (2).
- A RESERVED field set to zero.
- An Authentication Data field set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 20: Judgment #2

```
                                 1                   2                   3
             0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
            ! Next     44    !0!      0      ! Length          40            !              |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---           |
            !       0       !       0       ! Length          36            !  |           |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
            ! Number     1  ! Prot ID   3   ! SPI Size   4   ! Trans Cnt  3  !  |           |
            +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
            ! SPI value                                                     !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3       !       0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |SA Payload
          | ! Type  1  (EN) !       0       ! Transform ID     3     (3DES) !  | Proposal |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       3       !       0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  3  (IN) !       0       ! Transform ID     2     (SHA1) !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | !       0       !       0       ! Length           8            !  |           |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |           |
          | ! Type  5  (ESN)!       0       ! Transform ID     0      (No)  !  |           |
        --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 156 SA Payload contents

The NUT transmits an IKE_AUTH response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 157 SA Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

A Proposals field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 158 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero (last).
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 159 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including

Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.

- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 24: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 160 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 161 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix Y.
- A Ending Address field set to greater thatn or equal to Prefix Y.

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 162 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                    1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 163 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- An Ending Address field set to less than or equal to Prefix B.

Possible Problems:

- IKE_AUTH response has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDr, [CERT+],
AUTH,
[CP(CFG_REPLY)],
[N(IPCOMP_SUPPORTED)],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, TSi, TSr,
[N(ADDITIONAL_TS_POSSIBLE)],
[V+]
```

- Each of transforms can be located in the any order.

Test IKEv2.SGW.R.1.1.1.3: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key.

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   TH1             NUT             TN1             TH2
  (Host)          (SGW)           (SGW)           (Host)
    |               |               |               |
    |               |<--------------|               | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               |               |               | (Packet #1)
    |               |-------------->|               | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               |               |               | (Judgment #1)
    |               |               |               |
    |               |<--------------|               | IKE_AUTH request (HDR, SK {IDi, AUTH,
    |               |               |               |                            SAi2, TSi, TSr})
    |               |               |               | (Packet #2)
    |               |-------------->|               | IKE_AUTH response (HDR, SK {IDr, AUTH,
    |               |               |               |                             SAr2, TSi, TSr})
    |               |               |               | (Judgment #2)
    |               |               |               |
    |<--------------+===============+---------------| IPsec {Echo Request}
    |               |               |               | (Packet#3) (Judgment #3)
    |---------------+===============+-------------->| IPsec {Echo Reply}
    |               |               |               | (Packet #4) (Judgment #4)
    |               |               |               |
    V               V               V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1.

6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TH2 can send Echo Reply to TH1 instead of sending Echo Request.

Group 1.2. Use of Retransmission Timers

Test IKEv2.SGW.R.1.1.2.1: Receipt of retransmitted IKE_SA_INIT request

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT response, if a retransmission of the response is triggered.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 2.2 and 2.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |               * wait until retrans timer expires
   |-------X       | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #3)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #2)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #3)
   |               |
   V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1 (The Message ID is the same as Packet #1)

Part A: (BASIC)

1. TN1 starts to negotiate with TN1 by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. Observe the messages transmitted on Link A.
4. TN1 retransmits the same IKE_SA_INIT request as the message transmitted in Step 1 to the

NUT.

5. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 3: Judgment #2

The NUT never retransmits the same IKE_SA_INIT response as the response transmitted at Step 2.

Step 5: Judgment #3

The NUT transmits the same IKE_SA_INIT response as the response transmitted at Step 2.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.2.2: Receipt of retransmitted IKE_AUTH request

Purpose:

To verify an IKEv2 device transmits IKE_AUTH response, if a retransmission of the response is triggered.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT             TN1
(End-Node)      (End-Node)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |               | (Packet #2)
    |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |               | (Judgment #2)
    |               |
    |               * wait until retrans timer expires
    |-------X       | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |               | (Judgment #3)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |               | (Packet #3)
    |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |               | (Judgment #4)
    |               |
    V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #5 (The Message ID is the same as Packet #2)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. TN1 transmits an IKE_AUTH request to the NUT.

4. Observe the messages transmitted on Link A.
5. Observe the messages transmitted on Link A.
6. TN1 retransmits the same IKE_AUTH request as the request transmitted in Step 3 to the NUT.
7. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES","PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 5: Judgment #3
The NUT never retransmits the same IKE_AUTH response as the response transmitted at Step 4.

Step 7: Judgment #4
The NUT transmits the same IKE_AUTH response as the response transmitted at Step 4.

Possible Problems:

- None.

Group 1.3. State Synchronization and Connection Timeouts

Test IKEv2.SGW.R.1.1.3.1: State Synchronization with ICMP messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives ICMP messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1            NUT           TR1           TN1           TH2
(Host)         (SGW)        (Router)       (SGW)         (Host)
  |              |              |             |             |
  |              |<-------------+-------------|             | IKE_SA_INIT request
  |              |              |             |             |   (HDR, SAi1, KEi, Ni)
  |              |              |             |             | (Packet #1)
  |              |--------------+------------>|             | IKE_SA_INIT response
  |              |              |             |             |   (HDR, SAr1, KEr, Nr)
  |              |              |             |             | (Judgment #1)
  |              |              |             |             |
  |              |<-------------+-------------|             | IKE_AUTH request
  |              |              |             |             |  (HDR, SK {IDi, AUTH,SAi2, TSi, TSr})
  |              |              |             |             | (Packet #2)
  |              |--------------+------------>|             | IKE_AUTH Response
  |              |              |             |             |  (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |              |              |             |             | (Judgment #2)
  |              |              |             |             |
  |<-------------+==============+=============+-------------| IPsec {Echo Request}
  |              |              |             |             | (Packet #3) (Judgment #3)
  |--------------+==============+=============+------------>| IPsec {Echo Reply}
  |              |              |             |             | (Packet #4) (Judgment #4)
  |              |              |             |             |
  |              |<-------------|             |             | Destination Unreachable
  |              |              |             |             |  (No route to destination)
  |              |              |             |             | (Packet #5)
  |              |              |             |             |
  |<-------------+==============+=============+-------------| IPsec {Echo Request}
  |              |              |             |             | (Packet #6) (Judgment #5)
  |--------------+==============+=============+------------>| IPsec {Echo Reply}
  |              |              |             |             | (Packet #7) (Judgment #6)
  |              |              |             |             |
  V              V              V             V             V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See Common Packet #21
Packet #7	See Common Packet #25

- Packet #5: ICMPv6 Destination Unreachable

IPv6 Header	Source Address			TR1's Global Address on Link A
	Destination Address			NUT's Global Address on Link A
ICMPv6	Type			1
	Code			0
	Data	IP Header	Source Address	NUT's Global Address on Link A
			Destination Address	TN1's Global Address on Link X
			Next Header	50 (ESP)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1 and TN1 forwards the Echo Request with IPsec ESP using corresponding algorithms to the NUT.
6. Observe the messages transmitted on Link A..
7. After reception of an Echo Request from the NUT, TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..
9. TR1 transmit an ICMP Destination Unreachable Message to the NUT.
10. TH2 transmits an Echo Request to TH1 and TN1 forwards the Echo Request with IPsec ESP using corresponding algorithms to the NUT.
11. Observe the messages transmitted on Link A..
12. After reception of an Echo Request from the NUT, TH1 transmits an Echo Reply to TH2.
13. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT forwards an Echo Request.

Step 13: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.3.2: State Synchronization with IKE messages

Purpose:

To verify an IKEv2 device synchronizes its state when it receives IKE messages.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1           NUT           TR1           TN1           TH2
(Host)        (SGW)       (Router)        (SGW)        (Host)
  |             |             |             |             |
  |             |<------------+-------------|             | IKE_SA_INIT request
  |             |             |             |             |  (HDR, SAi1, KEi, Ni)
  |             |             |             |             | (Packet #1)
  |             |-------------+------------>|             | IKE_SA_INIT response
  |             |             |             |             |  (HDR, SAr1, KEr, Nr)
  |             |             |             |             | (Judgment #1)
  |             |             |             |             |
  |             |<------------+-------------|             | IKE_AUTH request
  |             |             |             |             |  (HDR, SK {IDi, AUTH,SAi2, TSi, TSr})
  |             |             |             |             | (Packet #2)
  |             |-------------+------------>|             | IKE_AUTH Response
  |             |             |             |             |  (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |             |             |             |             | (Judgment #2)
  |             |             |             |             |
  |<------------+=============+=============+-------------| IPsec {Echo Request}
  |             |             |             |             | (Packet #3) (Judgment #3)
  |-------------+=============+=============+------------>| IPsec {Echo Reply}
  |             |             |             |             | (Packet #4) (Judgment #4)
  |             |             |             |             |
  |             |<------------+-------------|             | IKE Message
  |             |             |             |             | (Packet #5)
  |             |             |             |             |
  |<------------+=============+=============+-------------| IPsec {Echo Request}
  |             |             |             |             | (Packet #6) (Judgment #5)
  |-------------+=============+=============+------------>| IPsec {Echo Reply}
  |             |             |             |             | (Packet #7) (Judgment #6)
  |             |             |             |             |
  V             V             V             V             V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Packet #5	See below
Packet #6	See Common Packet #21
Packet #7	See Common Packet #25

- Packet #5: cryptographicaly unprotected INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link A
	Destination Address	NUT's Global Address on Link X
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	41 (N)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	any
	Length	any
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	3 (ESP)
	SPI Size	0
	Notify Message Type	11 (INVALID_SPI)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1 and TN1 forwards the Echo Request with IPsec ESP using corresponding algorithms to the NUT.
6. Observe the messages transmitted on Link A..
7. After reception of an Echo Request from the NUT, TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..
9. TR1 transmit a cryptographically unprotected INFORMATIONAL request with Notify payload of type INVALID_ SPI to the NUT.
10. TH2 transmits an Echo Request to TH1 and TN1 forwards the Echo Request with IPsec ESP using corresponding algorithms to the NUT.
11. Observe the messages transmitted on Link A..
12. After reception of an Echo Request from the NUT, TH1 transmits an Echo Reply to TH2.
13. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 11: Judgment #5
The NUT forwards an Echo Request.

Step 13: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None

Test IKEv2.SGW.R.1.1.3.3: Close connections when receiving INITIAL_CONTACT

Purpose:

To verify an IKEv2 device closes connections when receiving INITIAL_CONTACT.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4
- [RFC 4718] - Sections 7.9

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Packet #1)
  |                  |----------------->|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Judgment #1)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |----------------->|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #3) (Judgment #3)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Packet #5)
  |                  |----------------->|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_AUTH request (HDR, SK {IDi, N(INITIAL_CONTACT),
  |                  |                  |                  |   AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #6)
  |                  |----------------->|                  | IKE_AUTH Response (HDR, SK {IDr,
  |                  |                  |                  |   AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Judgment #6)
  |                  |                  |                  |
  |         X<-------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #7) (Judgment #7)
  |------------------+=======X          |                  | IPsec {Echo Request}
  |                  |                  |                  | (Packet #8) (Judgment #8)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #9) (Judgment #9)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #10) (Judgment #10)
  |                  |                  |                  |
  V                  V                  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #1
Packet #6	See below
Packet #7	See Common Packet #21 This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 4.
Packet #8	See Common Packet #25 This packet is cryptographically protected by the CHILD_SA negotiated at Step 1 to Step 4.
Packet #9	See Common Packet #21 This packet is cryptographically

	protected by the CHILD_SA negotiated at Step 9 to Step 12.
Packet #10	See Common Packet #25 This packet is cryptographically protected by the CHILD_SA negotiated at Step 9 to Step 12.

- Packet #6: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Next Payload	41 (N)
	Other fields are same as the Common Packet #5	
N Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0 (undefined)
	SPI Size	0
	Notify Message Type	16384 (INITIAL_CONTACT)
AUTH Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Same as the Common Packet #5	
TSr Payload	Same as the Common Packet #5	

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1. TN1 forwards the Echo Request with IPsec ESP using corresponding algorithms to the NUT.
6. Observe the messages transmitted on Link A..
7. After reception of an Echo Request from the NUT, TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..
9. After reception of an Echo Reply from NUT, TN1 transmits IKE_SA_INIT request to the NUT.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request with a Notify payload of type INITIAL_CONTACT to the NUT.
12. Observe the messages transmitted on Link A..
13. TH2 transmits an Echo Request to TH1. TN1 forwards the Echo Request with IPsec ESP using the first negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link B...
15. TH1 transmits an Echo Request to TH1.
16. Observe the messages transmitted on Link A..
17. TH2 transmits an Echo Request to TH1. TN1 forwards the Echo Request with IPsec ESP using the second negotiated algorithms to the NUT.
18. Observe the messages transmitted on Link B...
19. TH1 transmits an Echo Request to TH1.
20. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #6
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 14: Judgment #7
The NUT never forwards an Echo Request to the TH1.

Step 16: Judgment #8
The NUT never forwards an Echo Request to the TH2 with IPsec ESP using the first negotiated algorithms or the second negotiated algorithms.

Step 18: Judgment #9
The NUT forwards an Echo Request.

Step 20: Judgment #10
The NUT forwards an Echo Reply with IPsec ESP using the second algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.3.4: Receiving Liveness check

Purpose:

To verify an IKEv2 device checks whether the other endpoint is alive.

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<---------------| INFORMATIONAL request (HDR, SK {})
  |                | (Packet #3)
  |--------------->| INFORMATIONAL response (HDR, SK {})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #17

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Test IKEv2.SGW.R.1.1.3.5: Receiving Delete Payload for IKE_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when IKE_SA is deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<--------------| INFORMATIONAL request (HDR, SK {D})
  |                | (Packet #3)
  |-------------->| INFORMATIONAL response (HDR, SK {})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0

	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	2
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Protocol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index	none

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits an INFORMATIONAL request with a Delete payload including 1 (IKE_SA) as Protocol ID, zero as SPI Size and no SPI value.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with no payloads.

Possible Problems:

- None

Test IKEv2.SGW.R.1.1.3.6: Receiving Delete Payload for CHILD_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Packet #2)
   |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Judgment #2)
   |               |
   |<--------------| INFORMATIONAL request (HDR, SK {D})
   |               | (Packet #3)
   |-------------->| INFORMATIONAL response (HDR, SK {D})
   |               | (Judgment #3)
   |               |
   V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0

	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	2
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits an INFORMATIONAL request with a Delete payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the TN1's inbound SPI value to be deleted as SPI value.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response with a Delete payload including 3 (ESP) as Protocol ID, 4 as SPI Size and the NUT's inbound SPI value to be deleted as SPI value.

Possible Problems:

- None

Group 1.4. Version Numbers and Forward Compatibility

Test IKEv2.SGW.R.1.1.4.1: Receipt of a higher minor version number

Purpose:

To verify an IKEv2 device drops a message with a higher minor version number and send a notification message.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  V                V
```

Packet #1	See below

- Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Other fields are same as the Common Packet #1	
	Major Version	2
	Minor Version	1
SA Payload	Same as the Common Packet #1	
KE Payload	Same as the Common Packet #1	
Ni, Nr Payload	Same as the Common Packet #1	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request with a higher minor version number.
2. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.4.2: Receipt of a higher major version number

Purpose:

To verify an IKEv2 device drops a message with a higher major version number and send a notification message.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT               TN1
 (SGW)             (SGW)
   |                 |
   |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 | (Packet #1)
   |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                 | (Judgment #1)
   |                 |
   |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                 | (Packet #2)
   |---------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                 | (Judgment #2)
   |                 |
   |<----------------| INFORMATIONAL request (HDR, SK{})
   |                 | (Packet #3)
   |-------X         | No Response
   |                 |   or
   |---------------->| INFORMATIONAL response (HDR, SK{N(INVALID_MAJOR_VERSION)}
   |                 | (Judgment #3)
   V                 V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: INFORMATIONAL response packet

IPv6 Header	Same as the common packet #17	
UDP Header	Same as the common packet #17	
IKEv2 Header	Other fields are same as the common packet #17	
	Major Version	3
	Minor Version	0
E Payload	Same as the common packet #17	

Part A: (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with a higher major version number to the NUT.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type INVALID_MAJOR_VERSION containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 164 Notify Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A SPI Size field set to zero.
- A Notify Message Type field set to INVALID_MAJOR_VERSION (5).
- A Notification Data field set to the highest version number it supports (2).

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.4.3: Unrecognized payload types and critical bit is not set

Purpose:

To verify an IKEv2 device ignores invalid payload types when the invalid type payload's critical bit is not set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Packet #2)
   |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Judgment #2)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK {P, N, SA, Ni, TSi, TSr})
   |               | (Packet #3)
   |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |               | (Judgment #3)
   |               |
   V               V

P: Payload with an invalid payload type
N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	All fields are same as Common Packet #15 Payload
UDP Header	All fields are same as Common Packet #15 Payload
IKEv2 Header	All fields are same as Common Packet #15 Payload

E Payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #15	
Invalid Payload	Next Payoad	41 (N)
	Critical	0
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #15 Payload	
SA Payload	All fields are same as Common Packet #15 Payload	
Ni, Nr Paylaod	All fields are same as Common Packet #15 Payload	
TSi Paylaod	All fields are same as Common Packet #15 Payload	
TSr Payload	All fields are same as Common Packet #15 Payload	

Part A: Invalid payload type 1 (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 1 and the invalid payload’s critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA’s SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Invalid payload type 32 (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 32 and the invalid payload’s critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA’s SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Invalid payload type 49 (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload type to the NUT. The E payload’s IKE Header Next Payload field is set to 49 and the invalid payload’s critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA’s SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Invalid payload type 255 (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a payload with invalid payload

type to the NUT. The E payload's IKE Header Next Payload field is set to 255 and the invalid payload's critical flag is not set. The request includes a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.4.4: Unrecognized payload types and critical bit is set

Purpose:

To verify an IKEv2 device ignores invalid payload types when the invalid type payload's critical bit is set.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT               TN1
 (SGW)             (SGW)
   |                 |
   |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 | (Packet #1)
   |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                 | (Judgment #1)
   |                 |
   |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                 | (Packet #2)
   |---------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                 | (Judgment #2)
   |                 |
   |<----------------| CREATE_CHILD_SA request (HDR, SK {P, N, SA, Ni, TSi, TSr})
   |                 | (Packet #3)
   |-------X         | No Response
   |                 |   or
   |---------------->| CREATE_CHILD_SA response (HDR, SK {N(UNSUPPORTED_CRITICAL_PAYLOAD)})
   |                 | (Judgment #3)
   |                 |
   V                 V

P: Payload with an invalid payload type
N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	All fields are same as Common Packet #15 Payload
UDP Header	All fields are same as Common Packet #15 Payload
IKEv2 Header	All fields are same as Common Packet #15 Payload

E Payload	Next Payload	*Invalid payload type value*
	Other fields are same as Common Packet #15	
Invalid Payload	Next Payoad	41 (N)
	Critical	1
	Reserved	0
	Payload Length	4
N Payload	All fields are same as Common Packet #15 Payload	
SA Payload	All fields are same as Common Packet #15 Payload	
Ni, Nr Paylaod	All fields are same as Common Packet #15 Payload	
TSi Paylaod	All fields are same as Common Packet #15 Payload	
TSr Payload	All fields are same as Common Packet #15 Payload	

Part A: Invalid payload type 1 and Critical bit is set (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 1 and the pointed pyaload's Critical bit is set.
6. Observe the messages transmitted on Link A..

Part B: Invalid payload type 32 and Critical bit is set (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A..
9. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 32 and the pointed pyaload's Critical bit is set.
12. Observe the messages transmitted on Link A..

Part C: Invalid payload type 49 and Critical bit is set (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A..
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A..
17. After reception of IKE_AUTH response from the NUT, TN1 transmits an CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 49 and the pointed pyaload's Critical bit is set.
18. Observe the messages transmitted on Link A..

Part D: Invalid payload type 255 and Critical bit is set (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A..
21. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A..
23. After reception of IKE_AUTH response from the NUT, TN1 transmits an

CREATE_CHILD_SA request including a payload invalid payload type to the NUT. The CREATE_CHILD_SA request's IKE Header Next Payload field is set to 255 and the pointed pyaload's Critical bit is set.

24. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (1).

Part B

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (32).

Part C

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (49).

Part D

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT doest not transmit any packets or transmits an INFORMATIONAL response with a Notify payload of type UNSUPPORTED_CRITICAL_PAYLOAD with the invalid payload type value (255).

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.4.5: Invalid Order Payloads

Purpose:

To verify an IKEv2 device properly handles IKE message with invalid order payloads.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(SGW)                (SGW)
  |                    |
  |<-------------------| IKE_SA_INIT request (HDR, SAi1, Ni, KEi)
  |                    | (Packet #1)
  |-------X            | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                    | (Judgment #1)
  |                    |
  V                    V
```

Packet #1	See Common Packet #1 KEi payload and Ni payload replace each other.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT never transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Group 1.5. Cookies

Test IKEv2.SGW.R.1.1.5.1: Cookies

Purpose:

To verify an IKEv2 device transmits IKE_SA_INIT response with a Notify payload of type COOKIE.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT               TN1
 (SGW)             (SGW)
   |                 |
   |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 | (Packet #1)
   |       ...       | ...
   |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                 | (Packet #2)
   |                 |
   |---------------->| IKE_SA_INIT response (HDR, N(COOKIE))
   |                 | (Judgment #1)
   |                 |
   |<----------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
   |                 | (Packet #3)
   |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                 | (Judgment #2)
   V                 V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1
Packet #3	See below

- Packet #3: IKE_SA_INIT request packet

IPv6 Header		Same as the common packet #1
UDP Header		Same as the common packet #1
IKEv2 Header		Other fields are same as the common packet #1
	Next Payload	41 (N)

N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: Notify payload of type Cookie Format (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT.
3. Observe the messages transmitted on Link A..
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE with the cookie data supplied by NUT
5. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 3: Judgment #1

The NUT transmits an IKE_SA_INIT response including a IKE Header which contains zero as IKE_SA Responder's SPI field and a Notify payload of type COOKIE containing following values:.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Protocol ID  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 165 Notify Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A SPI Size field set to zero.
- A Notify Message Type field set to COOKIE (16390).
- A Notification Data field set to the cookie data.

Step 5: Judgment #2

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.5.2: Invalid Cookies

Purpose:

To verify an IKEv2 device handles IKE_SA_INIT request with an invalid cookie data.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |      ...      | ...
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #2)
   |               |
   |-------------->| IKE_SA_INIT response (HDR, N(COOKIE))
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi, Ni)
   |               | (Packet #3)
   |-------------->| IKE_SA_INIT response (HDR, N(COOKIE"))
   |               | (Judgment #2)
   V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #1
Packet #3	See below

- Packet #3: IKE_SA_INIT request packet

IPv6 Header		Same as the common packet #1
UDP Header		Same as the common packet #1
IKEv2 Header		Other fields are same as the common packet #1
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0

	Notify Message Type	COOKIE (16390)
	Notification Data	The difference value than COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT.
3. Observe the messages transmitted on Link A..
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE with a cookie data unexpected by NUT.
5. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 3: Judgment #1
The NUT transmits an IKE_SA_INIT response including an IKE Header which contains zero as IKE_SA Responder's SPI field and a Notify payload of type COOKIE.

Step 5: Judgment #2
The NUT transmits an IKE_SA_INIT response including an IKE Header which contains zero as IKE_SA Responder’s SPI field and a Notify payload of type COOKIE with a new cookie data.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.5.3: Interaction of COOKIE and INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device handles interaction of COOKIE and INVALID_KE_PAYLOAD.

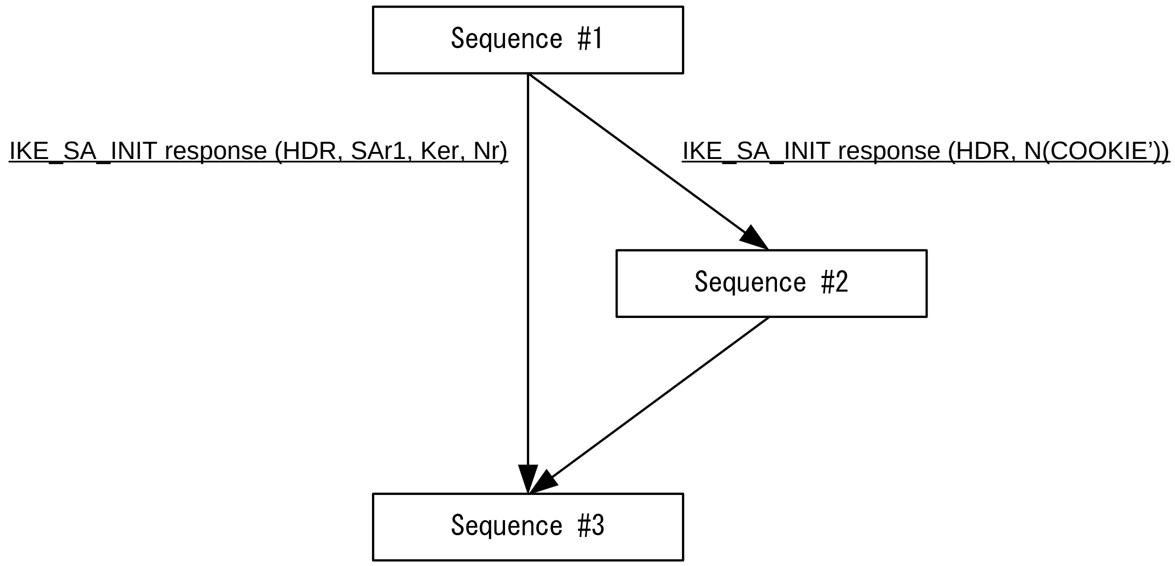
References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:



```
Sequence #1:
   NUT             TN1
  (SGW)           (SGW)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #1)
    |      ...      | ...
    |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |               | (Packet #2)
    |-------------->| IKE_SA_INIT response (HDR, N(COOKIE))
    |               | (Judgment #1)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
    |               | (Packet #3)
    |-------------->| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
    |               | (Judgment #2)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi ' , Ni)
    |               | (Packet #4)
    |------*1------>| IKE_SA_INIT response (HDR, N(COOKIE'))
    |               |   or
    |------*2------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #3)
    |               |
    If the NUT send IKE_SA_INIT response (HDR, N(COOKIE')), go to Sequence #2.
    If the NUT send IKE_SA_INIT response (HDR, SAr1, KEr, Nr), go to Sequence #3.
    Otherwise, this test is failed.

Sequence #2:
   NUT             TN1
  (SGW)           (SGW)
    |               |
    |<--------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KEi ' , Ni)
    |               | (Packet #5)
    |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |               | (Judgment #4)
    |               |
    Go to Sequence #3.

Sequence #3:
   NUT             TN1
  (SGW)           (SGW)
    |               |
    |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |               | (Packet #6)
    |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |               | (Judgment #5)
    |               |
    V               V
```

Packet #1	See below
Packet #2	See below
Packet #3	See below
Packet #4	See below
Packet #5	See below
Packet #6	See Common Packet #5

- Packet #1: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1
UDP Header	Same as the common packet #1

IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #2: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #3: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #4: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Same as the common packet #1	
Ni, Nr Payload	Same as the common packet #1	

- Packet #5: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)

	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT. The IKE_SA_INIT requests include an invalid KE payload which has different .DH Group # from proposing DH Group #.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has an invalid KE payload.
5. Observe the messages transmitted on Link A.
6. After reception of IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD, TN1 transmits an IKE_SA_INIT request with a valid KE payload.
7. Observe the messages transmitted on Link A.
8. If the IKE_SA_INIT response includes a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT request has a valid KE payload.
   A) Observe the messages transmitted on Link A
9. TN1 transmits an IKE_AUTH request.
10. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 5: Judgment #2
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type INVALID_KE_PAYLOAD.

Step 7: Judgment #3
The NUT transmits an IKE_SA_INIT response. The message can contain zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE. The message can containding "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 8A: Judgment #4
The message can containding "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #5

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.5.4: Interaction of COOKIE and INVALID_KE_PAYLOAD with unoptimized Initiator

Purpose:

To verify an IKEv2 device handles interaction of COOKIE and INVALID_KE_PAYLOAD.

References:

- [RFC 4306] - Sections 2.6 and 3.10.1
- [RFC 4718] - Sections 2.2, 2.4 and 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |       ...      | ...
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #2)
  |                |
  |--------------->| IKE_SA_INIT response (HDR, N(COOKIE))
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, N(COOKIE), SAi1, KEi, Ni)
  |                | (Packet #3)
  |--------------->| IKE_SA_INIT response (HDR, N(INVALID_KE_PAYLOAD))
  |                | (Judgment #2)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi', Ni)
  |                | (Packet #4)
  |--------------->| IKE_SA_INIT response (HDR, N(COOKIE'))
  |                | (Judgment #3)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, N(COOKIE'), SAi1, KE'i, Ni)
  |                | (Packet #5)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #4)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #6)
  |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #5)
  V                V
```

Packet #1	See below

Packet #2	See below
Packet #3	See below
Packet #4	See Common Packet #1
Packet #5	See below
Packet #6	See Common Packet #5

- Packet #1: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #2: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Same as the common packet #1	
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #3: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload	Same as the common packet #1	
KE Payload	Other fields are same as the common packet #1	
	DH Group #	14
Ni, Nr Payload	Same as the common packet #1	

- Packet #4: IKE_SA_INIT request packet

IPv6 Header	Same as the common packet #1	
UDP Header	Same as the common packet #1	
IKEv2 Header	Other fields are same as the common packet #1	
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)

	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

- Packet #5: IKE_SA_INIT request packet

IPv6 Header		Same as the common packet #1
UDP Header		Same as the common packet #1
IKEv2 Header		Other fields are same as the common packet #1
	Next Payload	41 (N)
N Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	Any
	Protocol ID	0
	SPI Size	0
	Notify Message Type	COOKIE (16390)
	Notification Data	The same value as COOKIE in IKE_SA_INIT response sent by NUT
SA Payload		Same as the common packet #1
KE Payload		Same as the common packet #1
Ni, Nr Payload		Same as the common packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. TN1 transmits a large number of IKE_SA_INIT requests to the NUT. The IKE_SA_INIT requests include an invalid KE payload which has different .DH Group # from proposing DH Group #.
3. Observe the messages transmitted on Link A.
4. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has an invalid KE payload.
5. Observe the messages transmitted on Link A.
6. After reception of IKE_SA_INIT response with a Notify payload of type INVALID_KE_PAYLOAD, TN1 transmits an IKE_SA_INIT request with a valid KE payload.
7. Observe the messages transmitted on Link A.
8. After reception of IKE_SA_INIT response with a Notify payload of type COOKIE, TN1 transmits an IKE_SA_INIT request which includes a Notify payload of type COOKIE. The IKE_SA_INIT still has a valid KE payload.
9. Observe the messages transmitted on Link A.
10. TN1 transtmits an IKE_AUTH request.
11. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 3: Judgment #1
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 5: Judgment #2
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type INVALID_KE_PAYLOAD.

Step 7: Judgment #3
The NUT transmits an IKE_SA_INIT response. The message contains zero as IKE_SA Responder's SPI field in IKE Header and a Notify payload of type COOKIE.

Step 9: Judgment #4
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 11: Judgment #5
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Possible Problems:

- None.
-

Group 1.6. Cryptographic Algorithm Negotiation

Test IKEv2.SGW.R.1.1.6.1: Cryptographic Algorithm Negotiation for IKE_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  From part A to part E, configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	***ENCR_AES_CBC***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	***ENCR_AES_CTR***	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	***PRF_AES128_CBC***	AUTH_HMAC_SHA1_96	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	***AUTH_AES_XCBC_96***	Group 2
Part E	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                 TN1
(SGW)               (SGW)
  |                   |
  |<------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   | (Packet #1)
  |------------------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                   | (Judgment #1)
  |                   |
  |<------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                   | (Packet #2)
  |------------------>| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                   | (Judgment #2)
  |                   |
  V                   V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload

as described above.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
4. Observe the messages transmitted on Link A..

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
8. Observe the messages transmitted on Link A..

Part C: PRF PRF_AES128_CBC (ADVANCED)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
12. Observe the messages transmitted on Link A..

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
14. Observe the messages transmitted on Link A..
15. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
16. Observe the messages transmitted on Link A..

Part E: D-H Group Group 14 (ADVANCED)

17. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
18. Observe the messages transmitted on Link A..
19. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request protected with the accepted proposal to the NUT.
20. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_AES_CBC”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_AES_CTR”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_AES128_CBC”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part D

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as accepted algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.2: Cryptographic Algorithm Negotiation for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-Shared key.

References:

- [RFC 4306] - Sections 2.7 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

  From part A to part F, TN1 transmits an IKE_AUTH request including a SA payload which contains the transforms as follows:

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**Extended Sequence Numbers**
Part A	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part B	**ENCR_AES_CTR**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part C	**ENCR_NULL**	AUTH_HMAC_SHA1_96	No Extended Sequence Numbers
Part D	ENCR_3DES	**AUTH_AES_XCBC_96**	No Extended Sequence Numbers
Part E	ENCR_3DES	**NONE**	No Extended Sequence Numbers
Part F	ENCR_3DES	AUTH_HMAC_SHA1_96	**Extended Sequence Numbers**

Procedure:

```
 TH1               NUT               TN1               TH2
(Host)            (SGW)             (SGW)             (Host)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 |                 |                 | (Packet #1)
  |                 |---------------->|                 | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 |                 |                 | (Judgment #1)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_AUTH request (HDR, SK {IDi, AUTH, N, SAi2,
  |                 |                 |                 |                             TSi, TSr})
  |                 |                 |                 | (Packet #2)
  |                 |---------------->|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, N, SAr2,
  |                 |                 |                 |                             TSi, TSr})
  |                 |                 |                 | (Judgment #2)
  |                 |                 |                 |
  |<----------------+=================+-----------------| IPsec {Echo Request}
  |                 |                 |                 | (Packet #3) (Judgment #3)
  |-----------------+=================+---------------->| IPsec {Echo Reply}
  |                 |                 |                 | (Packet #4) (Judgment #4)
  |                 |                 |                 |
  V                 V                 V                 V
```

Packet #1	See Common Packet #1

Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Part A: Encryption Algorithm ENCR_AES_CBC (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B...
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..

Part B: Encryption Algorithm ENCR_AES_CTR (ADVANCED)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
12. Observe the messages transmitted on Link A..
13. TH2 transmits an Echo Request to TH1.
14. Observe the messages transmitted on Link B...
15. TH1 transmits an Echo Reply to TH2.
16. Observe the messages transmitted on Link A..

Part C: Encryption Algorithm ENCR_NULL (ADVANCED)

17. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
18. Observe the messages transmitted on Link A..
19. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
20. Observe the messages transmitted on Link A..
21. TH2 transmits an Echo Request to TH1.
22. Observe the messages transmitted on Link B...
23. TH1 transmits an Echo Reply to TH2.
24. Observe the messages transmitted on Link A..

Part D: Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

25. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A..
27. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
28. Observe the messages transmitted on Link A..
29. TH2 transmits an Echo Request to TH1.
30. Observe the messages transmitted on Link B...
31. TH1 transmits an Echo Reply to TH2.
32. Observe the messages transmitted on Link A..

Part E: Integrity Algorithm NONE (ADVANCED)

33. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
34. Observe the messages transmitted on Link A..
35. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.

36. Observe the messages transmitted on Link A..
37. TH2 transmits an Echo Request to TH1.
38. Observe the messages transmitted on Link B...
39. TH1 transmits an Echo Reply to TH2.
40. Observe the messages transmitted on Link A..

Part F: Extended Sequence Numbers (ADVANCED)

41. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
42. Observe the messages transmitted on Link A..
43. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
44. Observe the messages transmitted on Link A..
45. TH2 transmits an Echo Request to TH1.
46. Observe the messages transmitted on Link B...
47. TH1 transmits an Echo Reply to TH2.
48. Observe the messages transmitted on Link A..

Part G: Security Protocol AH (ADVANCED)

49. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
50. Observe the messages transmitted on Link A..
51. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
52. Observe the messages transmitted on Link A..
53. TH2 transmits an Echo Request to TH1.
54. Observe the messages transmitted on Link B...
55. TH1 transmits an Echo Reply to TH2.
56. Observe the messages transmitted on Link A..

Part H: Security Protocol AH and Integrity Algorithm AUTH_AES_XCBC_96 (ADVANCED)

57. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
58. Observe the messages transmitted on Link A..
59. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
60. Observe the messages transmitted on Link A..
61. TH2 transmits an Echo Request to TH1.
62. Observe the messages transmitted on Link B...
63. TH1 transmits an Echo Reply to TH2.
64. Observe the messages transmitted on Link A..

Part I: Security Protocol AH and Extended Sequence Numbers (ADVANCED)

65. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
66. Observe the messages transmitted on Link A..
67. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request as described above to the NUT.
68. Observe the messages transmitted on Link A..
69. TH2 transmits an Echo Request to TH1.
70. Observe the messages transmitted on Link B...
71. TH1 transmits an Echo Reply to TH2.
72. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_AES_CBC”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part B

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_AES_CTR”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 14: Judgment #3
The NUT forwards an Echo Request.

Step 16: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part C

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 20: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_NULL”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 22: Judgment #3
The NUT forwards an Echo Request.

Step 24: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part D

Step 26: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 28: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 30: Judgment #3
The NUT forwards an Echo Request.

Step 32: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part E

Step 34: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 36: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “NONE” and “No Extended Sequence Numbers” as accepted algorithms.

Step 38: Judgment #3
The NUT forwards an Echo Request.

Step 40: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part F

Step 42: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 44: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “Extended Sequence Numbers” as accepted algorithms.

Step 46: Judgment #3
The NUT forwards an Echo Request.

Step 48: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part G

Step 50: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 52: Judgment #2
The NUT transmits an IKE_AUTH response including “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 54: Judgment #3

The NUT forwards an Echo Request.

Step 56: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part H

Step 58: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 60: Judgment #2
The NUT transmits an IKE_AUTH response including “AUTH_AES_XCBC_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 62: Judgment #3
The NUT forwards an Echo Request.

Step 64: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Part I

Step 66: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 68: Judgment #2
The NUT transmits an IKE_AUTH response including “AUTH_HMAC_SHA1_96” and “Extended Sequence Numbers” as accepted algorithms.

Step 70: Judgment #3
The NUT forwards an Echo Request.

Step 72: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.3: Receiving Multiple Transforms for IKE_SA

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with an multiple transforms payload.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  V                V
```

Packet #1	See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the transforms as follows:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_AES_CBC** **ENCR_3DES**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_AES128_CBC** **PRF_HMAC_SHA1**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96** **AUTH_HMAC_SHA1_96**	Group 2
Part D	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14** **Group 2**

- Packet #1 IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Same as the Common Packet #1	
SA Payload	Other fields are same as the common packet #1	
	SA Proposals	See SA Table below

KE Payload	Same as the Common Packet #1
Ni, Nr Payload	Same as the Common Packet #1

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A..

Part B: Multiple Pseudo-Random Functions (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link A..

Part C: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A..

Part D: Multiple D-H Groups (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.

8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.4: Receiving Multiple Proposals for IKE_SA

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with multiple proposals.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  V                V
```

Packet #1 See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the proposals as follows:

	IKE_SA_INIT exchanges Algorithms					
	Proposals	Protocol ID	Encryption	PRF	Integrity	D-H Group
Part A	Proposal #1	IKE	**ENCR_AES_CBC**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	Proposal #1	IKE	ENCR_3DES	**PRF_AES128_CBC**	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part C	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96**	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part D	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14**
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

- Packet #1 IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1
UDP Header	Same as the Common Packet #1
IKEv2 Header	Same as the Common Packet #1
SA Payload	Other fields are same as the common packet #1

	SA Proposals	See SA Table below
KE Payload	Same as the Common Packet #1	
Ni, Nr Payload	Same as the Common Packet #1	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		2
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0

			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
2. Observe the messages transmitted on Link A..

Part B: Multiple Pseudo-Random Functions (BASIC)

3. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
4. Observe the messages transmitted on Link A..

Part C: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
6. Observe the messages transmitted on Link A..

Part D: Multiple D-H Groups (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload as described above.
8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part C

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part D

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.5: Receiving Multiple Transforms for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with an unacceptable SA payload.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  | (Judgment #1)
  |                  |
  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  | (Packet #2)
  |----------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

From part A to part D, TN1 transmits an IKE_AUTH request including a SA payload which contains the transforms as follows:

	IKE_AUTH exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_AES_CBC** **ENCR_3DES**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_AES_XCBC_96** **AUTH_HMAC_SHA1_96**	No ESN
Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**ESN** **No ESN**

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5

UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Same as the Common Packet #5	
SA Payload	Other fields are Same as the Common Packet #5	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #5	
TSr Payload	Same as the Common Packet #5	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link A..

Part B: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
8. Observe the messages transmitted on Link A..

Part C: Multiple Extended Sequecnce Numbers (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.

10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.6: Receiving Multiple Proposals for CHILD_SA

Purpose:

To verify an IKEv2 device properly handles CHILD_SA request with an unacceptable SA payload.

References:

- [RFC 4306] - Sections 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  | (Judgment #1)
  |                  |
  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  | (Packet #2)
  |----------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below

TN1 transmits an IKE_AUTH request including a SA payload which contains the two proposals as follows:

	IKE_AUTH exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part B	Proposal #1	ESP	ENCR_3DES	**AUTH_AES_XCBC_96**	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part C	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	**ESN**
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5		
UDP Header	Same as the Common Packet #5		
IKEv2 Header	Same as the Common Packet #5		
E Payload	Same as the Common Packet #5		
IDi Payload	Same as the Common Packet #5		
AUTH Payload	Same as the Common Packet #5		
SA Payload	Other fields are Same as the Common Packet #5		
	SA Proposals	See below	
TSi Payload	Same as the Common Packet #5		
TSr Payload	Same as the Common Packet #5		

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		2
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8

			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
4. Observe the messages transmitted on Link A..

Part B: Multiple Integrity Algorithms (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
8. Observe the messages transmitted on Link A..

Part C: Multiple Extended Sequecnce Numbers (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including a SA payload as described above to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH response including a SA Proposal with “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.7: Sending INVALID_KE_PAYLOAD

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with an unacceptable SA payload.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Packet #2)
   |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Judgment #2)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi, TSr})
   |               | (Packet #3)
   |-------X       | CREATE_CHILD_SA response (HDR, SK {SA, Nr, KEr, TSi, TSr})
   |               |   or
   |-------------->| CREATE_CHILD_SA response (HDR, SK { N(INVALID_KE_PAYLOAD)})
   |               | (Judgment #3)
   V               V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request for rekeying CHILD_SA

IPv6 Header	Same as the Common Packet #15
UDP Header	Same as the Common Packet #15
IKEv2 Header	Same as the Common Packet #15
E Payload	Same as the Common Packet #15

N Payload	Same as the Common Packet #15	
SA Payload	Same as the Common Packet #15	
Ni, Nr Payload	Other fields are same as the Common Packet #15	
	Next Payload	34 (KE)
KEi Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	264
	DH Group #	14
	Reserved	0
	Key Exchange Data	any
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs. The CREATE_CHILD_SA contains a D-H Group transform to use D-H Group 2 and a Key Exchange payload which contains 14 (D-H Group 14) as DH Group # field and the Key Exchage Data.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits any packets or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_KE_PAYLOAD which contains 2 (D-H Group 2) as Notification Data.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.8: Sending INVALID_KE_PAYLOAD in Initial Exchange

Purpose:

To verify an IKEv2 device properly handles an invalid KE payload which has different D-H Group # from proposed D-H Group #.

References:

- [RFC 4306] - Sections 2.7, 3.4 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |<---------------| IKE_SA_INIT Request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT Response (HDR, N(INVALID_KE_PAYLOAD))
   |                | (Judgment #1)
   |                |
   |<---------------| IKE_SA_INIT Request (HDR, SAi1, KEi, Ni)
   |                | (Packet #2)
   |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                | (Judgment #2)
   |                |
   V                V
```

Packet #1	See below
Packet #2	See Common packet #1

- Packet #1: IKE_SA_INIT request

IPv6 Header	Same as the Common Packet #1	
UDP Header	Same as the Common Packet #1	
IKEv2 Header	Same as the Common Packet #1	
SA Payload	Same as the Common Packet #1	
KEi Payload	Next Payload	40 (Ni, Nr)
	Critical	0
	Reserved	0
	Payload Length	264
	DH Group #	14
	Reserved	0
	Key Exchange Data	any
Ni, Nr Payload	Same as the Common Packet #1	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request including a SA payload which contains a D-H Group transform proposes using D-H Group 2 and a Key Exchange payload which contains 14 (D-H Group 14) as DH Group # field and the Key Exchange Data.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including a Notify payload of type INVALID_KE_PAYLOAD which contains 2 (D-H Group 2) as Notification Data. The message's IKE_SA Responder's SPI value is set to zero.

Step 4: Judgment #2
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.6.9: Creating an IKE_SA without a CHILD_SA

Purpose:

To verify that an IKEv2 device can handles a failure of creating a CHILD_SA during the IKE_AUTH exchange.

References:

- [RFC 4718] - Sections 4.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  | (Judgment #1)
  |                  |
  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  | (Packet #2)
  |-------X          | IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  |<-----------------| INFORMATIONAL request (HDR, SK {})
  |                  | (Packet #3)
  |----------------->| INFORMATIONAL response (HDR, SK {})
  |                  | (Judgment #3)
  |                  |
  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #17

Packet #2: IKE_AUTH requet

Packet #2 is same as Common Packet #5 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type ENCR is replaced by the following SA Transfrom.

SA Transform	Next Payload	3 (more)
	Reserved	0

	Transform Length		8
	Transform Type		1 (ENCR)
	Reserved		0
	Transform ID		12 (AES_CBC)
	SA Attribute	Attribute Type	14 (Key Length)
		Attribute Value	128

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH request with unacceptable SA proposal for the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT never transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- Step 4
  The NUT can transmits an IKE_AUTH response with a Notify payload of type NO_PROPOSAL_CHOSEN.

Group 1.7. Traffic Selector Negotiation

Test IKEv2.SGW.R.1.1.7.1: Narrowing Traffic Selectors

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TH2	ANY	ANY	NUT	ANY	ANY
Outbound	NUT	ANY	ANY	TH2	ANY	ANY

The other packets are allowed to BYPASS IPsec protection.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2              TH3
(Host)           (SGW)            (SGW)            (Host)           (Host)
  |                |                |                |                |
  |                |<---------------|                |                | IKE_SA_INIT request (HDR, SAi1,
  |                |                |                |                |                      KEi, Ni)
  |                |                |                |                | (Packet #1)
  |                |--------------->|                |                | IKE_SA_INIT response (HDR, SAr1,
  |                |                |                |                |                       KEr, Nr)
  |                |                |                |                | (Judgment #1)
  |                |                |                |                |
  |                |<---------------|                |                | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                |                |                |                |                  SAi2, TSi, TSr})
  |                |                |                |                | (Packet #2)
  |                |--------------->|                |                | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |                |                |                |                |                   SAr2, TSi, TSr})
  |                |                |                |                | (Judgment #2)
  |                |                |                |                |
  |<---------------+================+----------------|                | IPsec {Echo Request}
  |                |                |                |                | (Packet #3) (Judgment #3)
  |----------------+================+--------------->|                | IPsec {Echo Reply}
  |                |                |                |                | (Packet #4) (Judgment #4)
  |                |                |                |                |
  |       X--------+================+---------------------------------| IPsec {Echo Request}
  |                |                |                |                | (Packet #5) (Judgment #5)
  |-------------------------------------------------------------------->| IPsec {Echo Reply}
  |                |                |                |                | (Packet #6) (Judgment #6)
  |                |                |                |                |
  V                V                V                V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below

- Packet #5: ICMPv6 Echo Request

IPv6 Header	Same as the Common Packet #21	
ESP	Same as the Common Packet #21	
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Same as the Common Packet #21	

- Packet #6: ICMPv6 Echo Request

IPv6 Header	Source Address	TH1's Global Address
	Destination Address	TH3's Global Address
ICMPv6 Header	Same as the Common Packet #25	

Part A (BASIC)

1. TN1 sends an IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. TN1 sends an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request packet to TH1.
6. Observe the messages transmitted on Link B...
7. TH1 transmits an Echo Reply packet to TH2.

8. Observe the messages transmitted on Link A..
9. TH3 transmits an Echo Request to TH1.
10. Observe the messages transmitted on Link B...
11. TH1 transmits an Echo Request to TH3.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms. The Traffic Selector is narrowed to allow only address range of TH2.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Request with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT never forwards an Echo Request.

Step 12: Judgment #6
The NUT forwards an Echo Request without IPsec ESP.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TH2 can send Echo Reply to TH1 instead of sending Echo Request.

Test IKEv2.SGW.R.1.1.7.2: TS_UNACCEPTABLE

Purpose:

To verify an IKEv2 device properly handles the Traffice Selector.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TH2	ANY	ANY	NUT	ANY	ANY
Outbound	NUT	ANY	ANY	TH2	ANY	ANY

The other packets are allowed to BYPASS IPsec protection.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Packet #2)
   |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Judgment #2)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi, TSr})
   |               | (Packet #3)
   |-------X       | CREATE_CHILD_SA response (HDR, SK {SA, Nr, KEr, TSi, TSr})
   |               |   or
   |-------------->| CREATE_CHILD_SA response (HDR, SK {N(TS_UNACCEPTABLE)})
   |               | (Judgment #3)
   V               V

N: REKEY_SA
```

Packet #1	See Common Packet #1

Packet #2	See below
Packet #3	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Same as the Common Packet #5	
N Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH2's Global Address on Link X
		Ending Address	TH2's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #9	
UDP Header	Same as the Common Packet #9	
IKEv2 Header	Same as the Common Packet #9	
E Payload	Same as the Common Packet #9	
SA Payload	Same as the Common Packet #9	
Ni, Nr Payload	Same as the Common Packet #9	
TSi Payload	Other fields are same as the Common Packet #9	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #9	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH3's Global Address on Link X
		Ending Address	TH3's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0

		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

Part A (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request including ICMPv6 (58) as IP Protocol ID value in Traffic Selector Payload.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits any packets or transmits a CREATE_CHILD_SA response including a Notify payload of type TS_UNACCEPTABLE.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.7.3: Narrowing Traffic Selectors

Purpose:

To verify an IKEv2 device allows the responder to choose a subset of the traffic proposed by the initiator.

References:

- [RFC4306] - Section 2.8
- [RFC4718] - Section 4.10

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration except Traffic Selector. Traffic Selector should be configured as following.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TH2	ANY	ANY	NUT	ANY	ANY
Outbound	NUT	ANY	ANY	TH2	ANY	ANY

The other packets are allowed to BYPASS IPsec protection.

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1             NUT             TN1             TH2             TH3
(Host)          (SGW)           (SGW)           (Host)          (Host)
  |               |               |               |               |
  |               |<--------------|               |               | IKE_SA_INIT request (HDR, SAi1,
  |               |               |               |               |                      KEi, Ni)
  |               |               |               |               | (Packet #1)
  |               |-------------->|               |               | IKE_SA_INIT response (HDR, SAr1,
  |               |               |               |               |                       KEr, Nr)
  |               |               |               |               | (Judgment #1)
  |               |               |               |               |
  |               |<--------------|               |               | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |               |               |               |               |                  SAi2, TSi, TSr})
  |               |               |               |               | (Packet #2)
  |               |-------------->|               |               | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |               |               |               |               |                   SAr2, TSi, TSr})
  |               |               |               |               | (Judgment #2)
  |               |               |               |               |
  |<--------------+===============+---------------|               | IPsec {Echo Request}
  |               |               |               |               | (Packet #3) (Judgment #3)
  |---------------+===============+-------------->|               | IPsec {Echo Reply}
  |               |               |               |               | (Packet #4) (Judgment #4)
  |               |               |               |               |
  |       X-------+===============+-------------------------------| IPsec {Echo Request}
  |               |               |               |               | (Packet #5) (Judgment #5)
  |-------------------------------------------------------------->| IPsec {Echo Reply}
  |               |               |               |               | (Packet #6) (Judgment #6)
  |               |               |               |               |
  V               V               V               V               V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH2's Global Address on Link X
		Ending Address	TH2's Global Address on Link X
	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40

		Start Port	0
		End Port	65535
		Starting Address	TH3's Global Address on Link X
		Ending Address	TH3's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH1's Global Address on Link A
		Ending Address	TH1's Global Address on Link A

- Packet #5: ICMPv6 Echo Request

IPv6 Header	Same as the Common Packet #21	
ESP	Same as the Common Packet #21	
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Same as the Common Packet #21	

- Packet #6: ICMPv6 Echo Request

IPv6 Header	Source Address	TH1's Global Address
	Destination Address	TH3's Global Address
ICMPv6 Header	Same as the Common Packet #25	

Part A (BASIC)

1. TN1 sends an IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. TN1 sends an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request packet to TH1.
6. Observe the messages transmitted on Link B...
7. TH1 transmits an Echo Reply packet to TH2.
8. Observe the messages transmitted on Link A..
9. TH3 transmits an Echo Request to TH1.
10. Observe the messages transmitted on Link B...
11. TH1 transmits an Echo Request to TH3.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms. The Traffic Selector is narrowed to allow the traffic from/to TH2.

Step 6: Judgment #3

The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Request with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT never forwards an Echo Request.

Step 12: Judgment #6
The NUT forwards an Echo Request without IPsec ESP.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TH2 can send Echo Reply to TH1 instead of sending Echo Request.

Group 1.8. Error Handling

Test IKEv2.SGW.R.1.1.8.1: INVALID_IKE_SPI

Purpose:

To verify an IKEv2 device properly handles IKE messages outside the context of IKE_SA.

References:

- [RFC 4306] - Sections 2.21

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr2, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, N+, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, N+, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi, TSr})
    |                | (Packet #3)
    |-------X        | CREATE_CHILD_SA response (HDR, SK {SA, Nr, KEr, TSi, TSr})
    |                |   or
    |--------------->| CREATE_CHILD_SA response (HDR, SK {N(INVALID_IKE_SPI)})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request (Part A)

IPv6 Header	Same as the Common Packet #15
UDP Header	Same as the Common Packet #15

IKEv2 Header		Other fields are same as the Common Packet #15
	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message plus 1
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message
E Payload		Same as the Common Packet #15
N Payload		Same as the Common Packet #15
SA Payload		Same as the Common Packet #15
Ni, Nr Payload		Same as the Common Packet #15
TSi Payload		Same as the Common Packet #15
TSr Payload		Same as the Common Packet #15

- Packet #3: CREATE_CHILD_SA request (Part A)

IPv6 Header		Same as the Common Packet #15
UDP Header		Same as the Common Packet #15
IKEv2 Header		Other fields are same as the Common Packet #15
	IKE_SA Initiator's SPI	The IKE_SA Initiator's SPI value used by this IKE message
	IKE_SA Responder's SPI	The IKE_SA Responder's SPI value used by this IKE message plus 1
E Payload		Same as the Common Packet #15
N Payload		Same as the Common Packet #15
SA Payload		Same as the Common Packet #15
Ni, Nr Payload		Same as the Common Packet #15
TSi Payload		Same as the Common Packet #15
TSr Payload		Same as the Common Packet #15

Part A: Different IKE_SA Initiator's SPI (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which contains different IKE_SA Initiator's SPI value from IKE_SA Initiator's SPI value in the IKE_AUTH request in Step 3.
6. Observe the messages transmitted on Link A..

Part B: Different IKE_SA Responder's SPI (ADVANCED)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A..
9. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which contains different IKE_SA Responder's SPI value from IKE_SA Responder's SPI value in the IKE_AUTH request in Step 3.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits any packets or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_IKE_SPI.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT does not transmits any packets or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_IKE_SPI.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.8.2: INVALID_SYNTAX

Purpose:

To verify an IKEv2 device properly handles IKE_SA_INIT request with an invalid syntax.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT response (HDR, SAr2, KEr, Nr)
   |                | (Judgment #1)
   |                |
   |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                | (Packet #2)
   |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                | (Judgment #2)
   |                |
   |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                | (Packet #3)
   |-------X        | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |                |   or
   |--------------->| CREATE_CHILD_SA response (HDR, SK {N(INVALID_SYNTAX)})
   |                | (Judgment #3)
   V                V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #15
UDP Header	Same as the Common Packet #15
IKEv2 Header	Same as the Common Packet #15
E Payload	Same as the Common Packet #15
N Payload	Same as the Common Packet #15
SA Payload	Same as the Common Packet #15

Ni, Nr Payload	Other fields are same as the common packet #15	
	Payload Length	4
	Nonce Data	Empty
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request which has no data as Nonce Data as Ni payload.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmits a CREATE_CHILD_SA response or transmits a CREATE_CHILD_SA response including a Notify payload of type INVALID_SYNTAX.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.8.3: INVALID_SELECTORS

Purpose:

To verify an IKEv2 device properly handles an ESP or AH packet whose selectors do not match those of the CHILD_SA.

References:

- [RFC 4306] - Sections 3.10.1
- [RFC 4307] - Sections 7.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |<---------------|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Packet #1)
  |                |--------------->|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Judgment #1)
  |                |                |                |
  |                |<---------------|                | IKE_AUTH request (HDR, SK {IDi, AUTH, ,
  |                |                |                |                     SAi2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |--------------->|                | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |                |                |                |                     SAi2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |                |                |
  |         X------+================+----------------| IPsec {Echo Request}
  |                |                |                | (Packet #3)
  |                |--------------->|                | INFORMATIONAL request (HDR, SK {N(INVALID_SELECTORS)}
  |                |                |                | (Judgment #3)
  V                V                V                V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #21

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5
UDP Header	Same as the Common Packet #5
IKEv2 Header	Same as the Common Packet #5
E Payload	Same as the Common Packet #5
IDi Payload	Same as the Common Packet #5
AUTH Payload	Same as the Common Packet #5

N Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH3's Global Address on Link X
		Ending Address	TH3's Global Address on Link X

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

Part A (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request as the above table to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TH2 transmits an Echo Request to TH1.
6. TN1 encapsulates an Echo Request with IPsec ESP usgin algorithms negotiated at between Step 1 and Step 5, though an Echo Request does not match the selector on TN1.
7. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL request with a Notify of type INVALID_SELECTORS.

Possible Problems:

- None.

Group 1.9 ID payload

Test IKEv2.SGW.R.1.1.9.1: Receiving IDi Type

Purpose:

To verify an IKEv2 device receives IDi payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDi payload which NUT receives from TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5
UDP Header	Same as the Common Packet #5
IKEv2 Header	Same as the Common Packet #5

E Payload	Same as the Common Packet #5	
IDi Payload	Other fields are same as the Common Packet #5	
	Payload Length	According to above configuration
	ID Type	According to above configuration
	Identification Data	According to above configuration
AUTH Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Same as the Common Packet #5	
TSr Payload	Same as the Common Packet #5	

Part A: ID_IPV4_ADDR (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
4. Observe the messages transmitted on Link A..

Part B: ID_FQDN (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
8. Observe the messages transmitted on Link A..

Part C: ID_RFC822_ADDR (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request including an IDi payload contains value specified as the above table to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.9.2: Sending IDr Type

Purpose:

To verify an IKEv2 device transmits IDr payload properly.

References:

- [RFC 4306] - Sections 3.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following configuration.

	IDr payload which NUT send to TN1	
	ID Type	**Idetification Data**
Part A	ID_IPV4_ADDR (1)	192.0.2.1
Part B	ID_FQDN (2)	example.com
Part C	ID_RFC822_ADDR (3)	jsmith@example.com

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Packet #2)
  |---------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  V                 V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A: ID_IPV4_ADDR (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..

Part B: ID_FQDN (BASIC)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
8. Observe the messages transmitted on Link A..

Part C: ID_RFC822_ADDR (BASIC)

9. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Part C

Step 10: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 12: Judgment #2
The NUT transmits an IKE_AUTH response including an IDr payload which contains value specified as the above table.

Possible Problems:

- None.

Group 1.10 Certificate payload

Test IKEv2.SGW.R.1.1.10.1: Sending Certificate Payload

Purpose:

To verify an IKEv2 device handles a CERTREQ payload and transmits a CERT payload propoerly.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Remote	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  | (Judgment #1)
  |                  |
  |<-----------------| IKE_AUTH request (HDR, SK {IDi, CERTREQ, AUTH, SAi2, TSi, TSr})
  |                  | (Packet #2)
  |----------------->| IKE_AUTH response (HDR, SK {IDr, CERT, AUTH, SAr2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5
UDP Header	Same as the Common Packet #5
IKEv2 Header	Same as the Common Packet #5
E Payload	Same as the Common Packet #5

IDi Payload	Next Payload	38 (CERTREQ)
	Oter fields are same as the Common Packet #5	
CERTREQ Payload	See below	
AUTH Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Same as the Common Packet #5	
TSr Payload	Same as the Common Packet #5	

CERTREQ Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate – Signature)
	Certificate Authority	any

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request with a CERTREQ payload to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response with a CERT payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding and the NUT’s certificate as Certificate Data.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.10.2: Sending Certificate Request Payload

Purpose:

To verify an IKEv2 device properly transmits CERTREQ payload.

References:

- [RFC 4306] - Sections 1.2 and 3.7

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)              (SGW)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr, CERTREQ)
  |                  | (Judgment #1)
  |                  |
  V                  V
```

Packet #1	See Common Packet #1

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response with a CERTREQ payload which contains 4 (X.509 Certificate - Signature) as Certificate Encoding.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.10.3: RSA Digital Signature

Purpose:

To verify an IKEv2 device authenticates the corresponding node by RSA Digital Signature.

References:

- [RFC 4306] - Sections 1.2 and 3.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Method
Local	X.509 Certificate - Signature

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1            NUT            TN1            TH2
(Host)         (SGW)          (SGW)          (Host)
  |              |              |              |
  |              |<-------------|              | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |              |              |              | (Packet #1)
  |              |------------->|              | IKE_SA_INIT response (HDR, SAr1, KEr, Nr, CERTREQ)
  |              |              |              | (Judgment #1)
  |              |              |              |
  |              |<-------------|              | IKE_AUTH request (HDR, SK {IDi, CERT, AUTH,
  |              |              |              |                           SAi2, TSi, TSr})
  |              |              |              | (Packet #2)
  |              |------------->|              | IKE_AUTH response (HDR, SK {IDr, AUTH,
  |              |              |              |                            SAr2, TSi, TSr})
  |              |              |              | (Judgment #2)
  |              |              |              |
  |<-------------+==============+--------------| IPsec {Echo Request}
  |              |              |              | (Packet#3) (Judgment #3)
  |--------------+==============+------------->| IPsec {Echo Reply}
  |              |              |              | (Packet #4) (Judgment #4)
  |              |              |              |
  V              V              V              V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #19

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5
UDP Header	Same as the Common Packet #5

IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Next Payload	37 (CERT5
	Oter fields are same as the Common Packet #5	
CERT Payload	See below	
AUTH Payload	Same as the Common Packet #5	
N Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Same as the Common Packet #5	
TSr Payload	Same as the Common Packet #5	

CERT Payload	Next Payload	39 (AUTH)
	Critical	0
	Reserved	0
	Payload Length	Any
	Certificate Encoding	4 (X.509 Certificate - Signature)
	Certificate Data	any

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request with a CERT payload to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using ENCR_3DES an AUTH_HMAC_SHA1_96.

Possible Problems:

- None.

Test IKEv2.EN.R.1.1.10.4: HEX string PSK

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key.

References:

- [RFC 4306] - Sections 2.15

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the following IKE peer configuration.

	Authentication Key Value
Local	0xabadcafeabadcafeabadcafeabadcafe (128 bit binary string)

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT             TN1
(SGW)           (SGW)
  |               |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |               | (Judgment #1)
  |               |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               | (Packet #2)
  |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               | (Judgment #2)
  |               |
  V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Group 1.11 Invalid values

Test IKEv2.SGW.R.1.1.11.1: Non zero RESERVED fields in IKE_SA_INIT request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                 TN1
 (SGW)               (SGW)
   |                   |
   |<------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                   | (Packet #1)
   |------------------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                   | (Judgment #1)
   |                   |
   V                   V
```

Packet #1	See Common Packet #1 All RESERVED fields are set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.11.2: Non zero RESERVED fields in IKE_AUTH request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5 All RESERVED fields are set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.11.3: Version bit is set

Purpose:

To verify an IKEv2 device ignores the content of Version in IKE messages.

References:

- [RFC 4306] - Sections 3.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                 TN1
(SGW)               (SGW)
  |                   |
  |<------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                   | (Packet #1)
  |------------------>| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                   | (Judgment #1)
  |                   |
  V                   V
```

Packet #1	See Common Packet #1 Version bit is set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request whose Version bit is set to one.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.11.4: Response bit is set

Purpose:

To verify an IKEv2 device ignores an IKE request message whose Response bit is set.

References:

- [RFC 4306] - Sections 2.21

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |-------X         | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  V                 V
```

Packet #1	See Common Packet #1 Response bit is set to one.

Part A (BASIC)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request whose Response bit is set to one.
2. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT never responds with an IKE_SA_INIT response to an IKE_SA_INIT request from the TN1.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.1.11.5: Unrecognized Notify Message Type

Purpose:

To verify an IKEv2 device ignores the unrecognized Notify Message Type in IKE messages.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni, N)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  V                V

N: Notify Payload with unrecognized Notify Message Type
```

Packet #1	See below

Packet #1: IKE_SA_INIT request

IPv6 Header	All fields are same as Common Packet #1	
UDP Header	All fields are same as Common Packet #1	
IKEv2 Header	All fields are same as Common Packet #1	
SA Payload	All fields are same as Common Packet #1	
KE Payload	All fields are same as Common Packet #1	
Ni, Nr paylaod	Next Payload	41 (Notify)
	Other fields are same as Common Packet #1	
N Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	8
	Procotol ID	0
	SPI Size	0
	Notify Message Type	See each part description.

Part A: Unrecognized Notify Message Type of error 16383 (BASIC)

5. TN starts to negotiate with NUT by sending IKE_SA_INIT request with a Notify payload of unrecognized Notify Message Type value.

6. Observe the messages transmitted on Link A.

Part B: Unrecognized Notify Message Type of status 65535 (BASIC)

7. TN starts to negotiate with NUT by sending IKE_SA_INIT request with a Notify payload of unrecognized Notify Message Type value.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Part B

Step 4: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Possible Problems:

- None.

Group 2. The CREATE_CHILD_SA Exchange

Group 2.1. Header and Payload Formats

Test IKEv2.SGW.R.1.2.1.1: Receipt of CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device transmits CREATE_CHILD_SA response using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.1.2,1.2 and 3.3.2
- [RFC 4307] - Sections 3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                | (Packet #3)
  |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                | (Judgment #3)
  |                |
  V                V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #15

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
6. Observe the messages transmitted on Link A..

Part B: Encrypted Payload Format (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A..
9. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
12. Observe the messages transmitted on Link A..

Part C: SA Payload Format (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A..
15. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A..
17. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
18. Observe the messages transmitted on Link A..

Part D: Nonce Payload Format (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A..
21. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A..
23. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
24. Observe the messages transmitted on Link A..

Part E: TSi Payload Format (BASIC)

25. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
26. Observe the messages transmitted on Link A..
27. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH request to the NUT.
28. Observe the messages transmitted on Link A..
29. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
30. Observe the messages transmitted on Link A..

Part F: TSr Payload Format (BASIC)

31. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
32. Observe the messages transmitted on Link A..
33. After a reception of IKE_SA_INIT response from the NUT, TN1 transmits IKE_AUTH

request to the NUT.

34. Observe the messages transmitted on Link A..
35. After reception of IKE_AUTH response from the NUT, TN1 transmits CREATE_CHILD_SA request to the NUT to rekey CHILD_SAs.
36. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 166 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to CREATE_CHILD_SA (36).
- A Flags field set to $(00000100)_2 = (4)_{10}$.
- A Message ID field set to the same value as corresponding IKEv2 request message's Message ID.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 8: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 167 Encrypted payload

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3

```
                               1                   2                   3
           0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
          ! Next    44    !0!      0      ! Length          40            !             |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---          |
          !       0       !        0      ! Length          36            !  |          |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
          ! Number     1  ! Prot ID    3  ! SPI Size  4   ! Trans Cnt  3  !  |          |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
          ! SPI value                                                     !  |          |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
         |!       3       !        0      ! Length           8            !  |          |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |SA Payload
         |! Type  1  (EN) !        0      ! Transform ID     3     (3DES) !  | Proposal |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
         |!       3       !        0      ! Length           8            !  |          |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
         |! Type  3  (IN) !        0      ! Transform ID     2     (SHA1) !  |          |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
         |!       0       !        0      ! Length           8            !  |          |
Transform |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |          |
         |! Type  5  (ESN)!        0      ! Transform ID     0       (No) !  |          |
      --- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 168 SA Payload contents

The NUT transmits a CREATE_CHILD_SA response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 169 SA Payload format

- A Next Payload field set to Nr Payload (40).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

A Proposals field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 170 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero (last).
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 171 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including

Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.
- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3
- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted Nonce Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                            Nonce Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 172 Nonce Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Nonce Data field set to random data generated by the transmitting entity. The size of the Nonce must between 16 and 256 octets.

Part E

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 30: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 173 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 174 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix Y.
- A Ending Address field set to greater thatn or equal to Prefix Y.

Part G

Step 32: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 34: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 36: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 175 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to the number of actual traffic selectors.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 176 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.

- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- An Ending Address field set to less than or equal to Prefix B.

Possible Problems:

- CREATE_CHILD_SA response has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
[N(IPCOMP_SUPPORTED)],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, Nr, [KEr], TSi, TSr,
[N(ADDITIONAL_TS_POSSIBLE)]
```

- Each of transforms can be located in the any order.

Group 2.2. Use of Retransmission Timers

Test IKEv2.SGW.R.1.2.2.1: Receipt of CREATE_CHILD_SA requests

Purpose:

To verify an IKEv2 device retransmits CREATE_CHILD_SA request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 2.1, 2.2 and 2.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                TN1
 (SGW)              (SGW)
   |                  |
   |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                  | (Packet #1)
   |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                  | (Judgment #1)
   |                  |
   |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                  | (Packet #2)
   |----------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                  | (Judgment #2)
   |                  |
   |<-----------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                  | (Packet #3)
   |----------------->| CREATE_CHILD_SA Rsponce Catl (HDR, SK {SA, Nr, TSi, TSr})
   |                  | (Judgment #3)
   |                  |
   |                  * wait until retrans timer expires
   |----------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |                  | (Judgment #4)
   |                  |
   |<-----------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                  | (Packet #4)
   |----------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |                  | (Judgment #5)
   |                  |
   V                  V

 N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #15
Packet #4	See Common Packet #15 (same Message ID as Pcket #3)

Part A: (BASIC)

1. TN1 transmits IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 trasmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits CREATE_CHILD_SA request.
6. Observe the messages transmitted on Link A.
7. Observe the messages transmitted on Link A..
8. TN1 transmits the same CREATE_CHILD_SA request packet as Step 5.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 7: Judgment #4
The NUT never retransmits a CREATE_CHILD_SA response which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Step 9: Judgment #5
The NUT retransmits a CREATE_CHILD_SA response which has the same Message ID value as the previous CREATE_CHILD_SA request’s Message ID value in IKE Header.

Possible Problems:

- none

Group 2.3. State Synchronization and Connection Timeouts

Test IKEv2.SGW.R.1.2.3.1: Receiving Delete Payload for Multiple CHILD_SA

Purpose:

To verify an IKEv2 device transmits a Delete Payload, when CHILD_SAs are deleted.

References:

- [RFC 4306] - Sections 2.4 and 3.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Packet #2)
  |-------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni, TSi, TSr})
  |                 | (Packet #3)
  |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                 | (Judgment #3)
  |                 |
  |<--------------| INFORMATIONAL request (HDR, SK {D})
  |                 | (Packet #4)
  |-------------->| INFORMATIONAL request (HDR, SK {D})
  |                 | (Judgment #4)
  |                 |
  V                 V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Same as the Common Packet #5	
N Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	6 (TCP)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #9	
UDP Header	Same as the Common Packet #9	
IKEv2 Header	Same as the Common Packet #9	
E Payload	Same as the Common Packet #9	
N Payload	Same as the Common Packet #9	
SA Payload	Same as the Common Packet #9	
Ni, Nr Payload	Same as the Common Packet #9	
TSi Payload	Other fields are same as the Common Packet #9	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #9	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix Y:0000:0000:0000:0000
		Ending Address	Prefix Y:ffff:ffff:ffff:ffff

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	58 (ICMPv6)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	Prefix B:0000:0000:0000:0000
		Ending Address	Prefix B:ffff:ffff:ffff:ffff

- Packet #4: INFORMATIONAL request

IPv6 Header		Same as the Common Packet #17
UDP Header		Same as the Common Packet #17
IKEv2 Header		Same as the Common Packet #17
E Payload		Other fields are same as the Common Packet #17
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	16
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	2
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange
		SPI negotiated by CREATE_CHILD_SA exchange

Part A: (ADVANCED)

1. TN starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits a CREATE_CHILD_SA request to establish a new CHILD_SA to the NUT.
6. Observe the messages transmitted on Link A..
7. TN1 transmits an INFORMATIONAL request with a Delete payload including the first negotiated CHILD_SA's inbound SPI and the second negotiated CHILD_SA's inbound SPI.
8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 8: Judgment #4
The NUT transmits an INFORMATIONAL response with delete payload for SPIs which are negotiated by Initial Exchange and CREATE_CHILD_SA exchange.

Possible Problems:

- INFORMATIONAL response from NUT may not contain Delete Payload by implementation policy. This behavior is defined at section 1.4 in RFC 4306 as an exception.

Group 2.4. Cryptographic Algorithm Negotiation

Test IKEv2.SGW.R.1.2.4.1: Sending NO_PROPOSAL_CHOSEN

Purpose:

To verify an IKEv2 device properly handles an IKE_AUTH request with an unacceptable SA payload.

References:

- [RFC 4306] - Sections 2.7 and 3.10.1
- [RFC 4718] - Sections 2.1 and 2.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |---------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  |<----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Packet #2)
  |---------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  |<----------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                 | (Packet #3)
  |-------X         | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                 | (Judgment #3)
  |                 |   or
  |---------------->| CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN)})
  |                 | (Judgment #3)
  |                 |
  V                 V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #15	
UDP Header	Same as the Common Packet #15	
IKEv2 Header	Same as the Common Packet #15	
E Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
SA Payload	Other fields are same as the Common Packet #15	
	SA Proposals	See below
Ni, Nr Payload	Same as the Common Packet #15	
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		36
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		3
		SPI		any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	12 (AES_CBC)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	5 (AES_XCBC_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	1 (ESN)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 trasmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits a CREATE_CHILD_SA request to rekey the established CHILD_SAs to the NUT. The CREATE_CHILD_SA request includes a SA payload with a proposal unaccepted by the NUT.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT does not transmit a CREATE_CHILD_SA response or transmits a CREATE_CHILD_SA response including a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- None.

Group 2.5. Rekeying CHILD_SA Using a CREATE_CHILD_SA exchange

Test IKEv2.SGW.R.1.2.5.1: Close the replaced CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the CREATE_CHILD_SA Exchanges to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Packet #1)
  |                  |----------------->|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Judgment #1)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |----------------->|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #3) (Judgment #3)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |                  |                  |                  |
  |                  |<-----------------|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Packet #5)
  |                  |----------------->|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |                  |<-----------------|                  | INFORMATIONAL request (HDR, SK {D})
  |                  |                  |                  | (Packet #6)
  |                  |----------------->|                  | INFORMATIONAL response (HDR, SK {D})
  |                  |                  |                  | (Judgment #6)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See below

- Packet #6: INFORMATIONAL request

IPv6 Header		Same as the Common Packet #17
UDP Header		Same as the Common Packet #17
IKEv2 Header		Same as the Common Packet #17
E Payload		Other fields are same as the Common Packet #17
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B...
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..
9. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
10. Observe the messages transmitted on Link A..
11. TN1 transmits an INFORMATIONAL request including a Delete payload with the old CHILD_SA's SPI value to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3

The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 12: Judgment #6
The NUT transmits an INFORMATIONAL response including a Delete payload with the old CHILD_SA’s SPI value to the TN1.

Possible Problems:

- None.

Test IKEv2.SGW.R.1.2.5.2: Use of the new CHILD_SA

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of CHILD_SAs.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1             NUT            TN1            TH2
(Host)          (SGW)          (SGW)          (Host)
  |               |              |              |
  |               |<-------------|              | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               |              |              | (Packet #1)
  |               |------------->|              | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |               |              |              | (Judgment #1)
  |               |              |              |
  |               |<-------------|              | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               |              |              | (Packet #2)
  |               |------------->|              | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               |              |              | (Judgment #2)
  |               |              |              |
  |<--------------+==============+--------------| IPsec {Echo Request}
  |               |              |              | (Packet #3) (Judgment #3)
  |---------------+==============+------------->| IPsec {Echo Reply}
  |               |              |              | (Packet #4) (Judgment #4)
  |               |              |              |
  |               |<-------------|              | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |               |              |              | (Packet #5)
  |               |------------->|              | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |               |              |              | (Judgment #5)
  |               |              |              |
  |               |<-------------|              | INFORMATIONAL request (HDR, SK {D})
  |               |              |              | (Packet #6)
  |               |--------------|              | INFORMATIONAL response (HDR, SK {D})
  |               |              |              | (Judgment #6)
  |               |              |              |
  |<--------------+==============+--------------| IPsec {Echo Request}  (new CHILD_SA)
  |               |              |              | (Packet #7) (Judgment #7)
  |---------------+==============+------------->| IPsec {Echo Reply}    (new CHILD_SA)
  |               |              |              | (Packet #8) (Judgment #8)
  |               |              |              |
  V               V              V              V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See below
Packet #7	See Common Packet #21 (encrypted by the new CHILD_SA)
Packet #8	See Common Packet #25

Packet #6: INFORMATIONAL request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
UDP Header	Source Port	500
	Destination Port	500
IKEv2 Header	IKE_SA Initiator's SPI	any
	IKE_SA Responder's SPI	any
	Next Payload	46 (E)
	Major Version	2
	Minor Version	0
	Exchange Type	37 (INFORMATIONAL)
	X (bits 0-2 of Flags)	0
	I (bit 3 of Flags)	any
	V (bit 4 of Flags)	0
	R (bit 5 of Flags)	0
	X (bits 6-7 Flags)	0
	Message ID	0
	Length	any
E Payload	Next Payload	42 (D)
	Critical	0
	Reserved	0
	Payload Length	any
	Initialization Vector	The same value as block length of the underlying encryption algorithm
	Encrypted IKE Payloads	Subsequent payloads encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Integrity Checksum Data	The Cryptographic checksum of the entire message
D Payload	Next Payload	0
	Critical	0
	Reserved	0
	Payload Length	12
	Protocol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index	NUT's inbound CHILD_SA SPI value to be deleted

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request using the first negotiated algorithms to the NUT.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.

8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with a Delete payload to the NUT.
12. Observe the messages transmitted on Link A.
13. TH2 transmits an Echo Request to TH1. TH1. TN1 forwards an Echo Request using the second negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Reply to TH2.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 12: Judgment #6
The NUT transmits an INFORMATIONAL response with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the NUT's inbound SPI value to be deleted as SPI value.

Step 14: Judgment #7
The NUT forwards an Echo Request.

Step 16: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.5.3: Receiving Multiple Transform

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)            (SGW)
   |                |
   |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                | (Judgment #1)
   |                |
   |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                | (Packet #2)
   |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                | (Judgment #2)
   |                |
   |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                | (Packet #3)
   |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |                | (Judgment #3)
   V                V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

From part A to part C, TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the transforms as follows:

	CREATE_CHILD_SA exchanges Algorithms		
	Encryption	**Integrity**	**ESN**
Part A	**ENCR_3DES** **ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
Part B	ENCR_3DES	**AUTH_HMAC_SHA1_96** **AUTH_AES_XCBC_96**	No ESN

Part C	ENCR_3DES	AUTH_HMAC_SHA1_96	**No ESN ESN**

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #15	
UDP Header	Same as the Common Packet #15	
IKEv2 Header	Same as the Common Packet #15	
E Payload	Same as the Common Packet #15	
IDi Payload	Same as the Common Packet #15	
AUTH Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
SA Payload	Other fields are same as the Common Packet #15	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

6. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.5.4: Receiving Multiple Proposal

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transforms to rekey CHILD_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT                TN1
 (SGW)              (SGW)
   |                  |
   |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                  | (Packet #1)
   |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                  | (Judgment #1)
   |                  |
   |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                  | (Packet #2)
   |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                  | (Judgment #2)
   |                  |
   |<-----------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
   |                  | (Packet #3)
   |----------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
   |                  | (Judgment #3)
   V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the two proposals as follows:

	CREATE_CHILD_SA exchanges Algorithms				
	Proposal	**Protocol ID**	**Encryption**	**Integrity**	**ESN**
Part A	Proposal #1	ESP	**ENCR_AES_CBC**	AUTH_HMAC_SHA1_96	No ESN
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part B	Proposal #1	ESP	ENCR_3DES	**AUTH_AES_XCBC_96**	No ESN

	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN
Part C	Proposal #1	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	**ESN**
	Proposal #2	ESP	ENCR_3DES	AUTH_HMAC_SHA1_96	No ESN

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #15	
UDP Header	Same as the Common Packet #15	
IKEv2 Header	Same as the Common Packet #15	
E Payload	Same as the Common Packet #15	
IDi Payload	Same as the Common Packet #15	
AUTH Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
SA Payload	Other fields are same as the Common Packet #15	
	SA Proposals	See below
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		40
		Proposal #		1
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		40
		Proposal #		2
		Proposal ID		3 (ESP)
		SPI Size		4
		# of Transforms		4
		SPI		Any
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8

			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	5 (ESN)
			Reserved	0
			Transform ID	0 (No ESN)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Integrity Algorithms (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Extended Sequecnce Numbers (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.5.5: Perfect Forward Secrecy

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA exchange when Perfect Forward Secrecy enables.

References:

- [RFC 4306] - Sections 2.12

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1               NUT               TN1               TH2
(Host)            (SGW)             (SGW)             (Host)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 |                 |                 | (Packet #1)
  |                 |---------------->|                 | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 |                 |                 | (Judgment #1)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 |                 |                 | (Packet #2)
  |                 |---------------->|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 |                 |                 | (Judgment #2)
  |                 |                 |                 |
  |<----------------+=================+-----------------| IPsec {Echo Request}
  |                 |                 |                 | (Packet #3) (Judgment #3)
  |-----------------+=================+---------------->| IPsec {Echo Reply}
  |                 |                 |                 | (Packet #4) (Judgment #4)
  |                 |                 |                 |
  |                 |<----------------|                 | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, KEi, TSi,
TSr})
  |                 |                 |                 | (Packet #5)
  |                 |---------------->|                 | CREATE_CHILD_SA response (HDR, SK {SA, Nr, KEr, TSi,
TSr})
  |                 |                 |                 | (Judgment #5)
  |                 |                 |                 |
  |                 |<----------------|                 | INFORMATIONAL request (HDR, SK {D})
  |                 |                 |                 | (Packet #6)
  |                 |-----------------|                 | INFORMATIONAL response (HDR, SK {D})
  |                 |                 |                 | (Judgment #6)
  |                 |                 |                 |
  |<----------------+=================+-----------------| IPsec {Echo Request}  (new CHILD_SA)
  |                 |                 |                 | (Packet #7) (Judgment #7)
  |-----------------+=================+---------------->| IPsec {Echo Reply}    (new CHILD_SA)
  |                 |                 |                 | (Packet #8) (Judgment #8)
  |                 |                 |                 |
  V                 V                 V                 V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below
Packet #7	See Common Packet #21 (encrypted by the new CHILD_SA)
Packet #8	See Common Packet #25

- Packet #5: CREATE_CHILD_SA response

IPv6 Header	Same as the Common Packet #15	
UDP Header	Same as the Common Packet #15	
IKEv2 Header	Same as the Common Packet #15	
E Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
N Payload	Same as the Common Packet #15	
SA Payload	Same as the Common Packet #15	
Ni Payload	Next Payload	34 (KE)

KEi Payload	Next Payload	44 (TSi)
	Critical	0
	Reserved	0
	Payload Length	136
	DH Group #	2
	Reserved	0
	Key Exchange Data	any
TSi Payload	Same as the Common Packet #15	
TSr Payload	Same as the Common Packet #15	

- Packet #6: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	12
	Procotol ID	3 (ESP)
	SPI Size	4
	# of SPIs	1
	Security Parameter Index(es) (SPI)	SPI negotiated by Initial Exchange

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request using the first negotiated algorithms to the NUT.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with a Delete payload to the NUT.
12. Observe the messages transmitted on Link A.
13. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request using the second negotiated algorithms to the NUT.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Reply to TH2.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 12: Judgment #6
The NUT transmits an INFORMATIONAL response with a Delete payload. The Delete payload includes 3 (ESP) as Protocol ID, 4 as SPI Size and the NUT’s inbound SPI value to be deleted as SPI value.

Step 14: Judgment #7
The NUT forwards an Echo Request.

Step 16: Judgment #8
The NUT forwards an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.5.6: Use of the old CHILD_SA

Purpose:

To verify an IKEv2 device properly handles new CHILD_SA and old CHILD_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Packet #1)
  |                  |----------------->|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Judgment #1)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |----------------->|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #3) (Judgment #3)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |                  |                  |                  |
  |                  |<-----------------|                  | CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
  |                  |                  |                  | (Packet #5)
  |                  |----------------->|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}  (old CHILD_SA)
  |                  |                  |                  | (Packet #6) (Judgment #6)
  |------------------+==================+----------------->| IPsec {Echo Reply}    (old CHILD_SA)
  |                  |                  |                  | (Packet #7) (Judgment #7)
  |                  |                  |                  |
  V                  V                  V                  V

N: REKEY_SA
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #3
Packet #3	See Common Packet #21

Packet #4	See Common Packet #25
Packet #5	See Common Packet #15
Packet #6	See Common Packet #21 (encrypted by the old CHILD_SA)
Packet #7	See Common Packet #25

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request using the first negotiated algorithms to the NUT.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
10. Observe the messages transmitted on Link A.
11. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request using the first negotiated algorithms again.
12. Observe the messages transmitted on Link B.
13. TH1 transmits an Echo Reply to TH2.
14. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 12: Judgment #6
The NUT forwards an Echo Request.

Step 14: Judgment #7

The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Possible Problems:

- none

Group 2.6. Rekeying IKE_SAs Using a CREATE_CHILD_SA exchange

Test IKEv2.SGW.R.1.2.6.1: Sending CREATE_CHILD_SA response

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8 and 2.18

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1                TH2
(Host)             (SGW)              (SGW)              (Host)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |                  | (Packet #1)
  |                  |----------------->|                  | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |                  | (Judgment #1)
  |                  |                  |                  |
  |                  |<-----------------|                  | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  |                  | (Packet #2)
  |                  |----------------->|                  | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  |                  | (Judgment #2)
  |                  |                  |                  |
  |<-----------------+==================+------------------| IPsec {Echo Request}
  |                  |                  |                  | (Packet #3) (Judgment #3)
  |------------------+==================+----------------->| IPsec {Echo Reply}
  |                  |                  |                  | (Packet #4) (Judgment #4)
  |                  |                  |                  |
  |                  |<-----------------|                  | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                  |                  |                  | (Packet #5)
  |                  |----------------->|                  | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                  |                  |                  | (Judgment #5)
  |                  |                  |                  |
  V                  V                  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #11

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B...
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A..
9. TN1 transmits a CREATE_CHILD_SA request including a SA payload. The proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator’s SPI value.
10. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including“ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the proposal in the SA payload Response includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA Responder’s SPI value in the SPI field.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.2: Receipt of cryptographically valid message on the old SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1              NUT              TN1              TH2
(Host)           (SGW)            (SGW)            (Host)
  |                |                |                |
  |                |<---------------|                | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                |                |                | (Packet #1)
  |                |--------------->|                | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                |                |                | (Judgment #1)
  |                |                |                |
  |                |<---------------|                | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                |                |                | (Packet #2)
  |                |--------------->|                | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |                |                | (Judgment #2)
  |                |                |                |
  |<---------------+================+----------------| IPsec {Echo Request}
  |                |                |                | (Packet #3) (Judgment #3)
  |----------------+================+--------------->| IPsec {Echo Reply}
  |                |                |                | (Packet #4) (Judgment #4)
  |                |                |                |
  |                |<---------------|                | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                |                |                | (Packet #5)
  |                |--------------->|                | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                |                |                | (Judgment #5)
  |                |                |                |
  |                |<---------------|                | INFORMATION Request (HDR, SK {})
  |                |                |                | (Packet #6)
  |                |--------------->|                | INFORMATIONAL response (HDR, SK {})
  |                |                |                | (Judgment #6)
  |                |                |                |
  V                V                V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Packet #5	See Common Packet #11
Packet #6	See Common Packet #17 (encrypted by the old IKE_SA)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with no payloads protected by the old IKE_SA.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the proposal in the SA payload includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA Responder's SPI value in the SPI field.

Step 12: Judgment #6
The NUT responds with an INFORMATIONAL response with no payloads protected by the old IKE_SA.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.3: Receipt of cryptographically valid message on the new SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1            NUT           TN1           TH2
(Host)         (SGW)         (SGW)         (Host)
  |              |             |             |
  |              |<------------|             | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |              |             |             | (Packet #1)
  |              |------------>|             | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |              |             |             | (Judgment #1)
  |              |             |             |
  |              |<------------|             | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |              |             |             | (Packet #2)
  |              |------------>|             | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |              |             |             | (Judgment #2)
  |              |             |             |
  |<-------------+=============+-------------| IPsec {Echo Request}
  |              |             |             | (Packet #3) (Judgment #3)
  |--------------+=============+------------>| IPsec {Echo Reply}
  |              |             |             | (Packet #4) (Judgment #4)
  |              |             |             |
  |              |<------------|             | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |              |             |             | (Packet #5)
  |              |------------>|             | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |              |             |             | (Judgment #5)
  |              |             |             |
  |              |<------------|             | INFORMATION Request (HDR, SK {})
  |              |             |             | (Packet #6)
  |              |------------>|             | INFORMATIONAL response (HDR, SK {})
  |              |             |             | (Judgment #6)
  |              |             |             |
  V              V             V             V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25

Packet #5	See Common Packet #11
Packet #6	See Common Packet #17 (encrypted by the new IKE_SA)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.
12. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms. And the proposal in the SA payload includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA Responder's SPI value in the SPI field.

Step 12: Judgment #6
The NUT responds with an INFORMATIONAL response with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.4: Close the replaced IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.8 and 5.11

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1               NUT               TN1               TH2
(Host)            (SGW)             (SGW)             (Host)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 |                 |                 | (Packet #1)
  |                 |---------------->|                 | IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 |                 |                 | (Judgment #1)
  |                 |                 |                 |
  |                 |<----------------|                 | IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 |                 |                 | (Packet #2)
  |                 |---------------->|                 | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 |                 |                 | (Judgment #2)
  |                 |                 |                 |
  |<----------------+=================+-----------------| IPsec {Echo Request}
  |                 |                 |                 | (Packet #3) (Judgment #3)
  |-----------------+=================+---------------->| IPsec {Echo Reply}
  |                 |                 |                 | (Packet #4) (Judgment #4)
  |                 |                 |                 |
  |                 |<----------------|                 | CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                 |                 |                 | (Packet #5)
  |                 |---------------->|                 | CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                 |                 |                 | (Judgment #5)
  |                 |                 |                 |
  |                 |<----------------|                 | INFORMATION Request (HDR, SK {D})
  |                 |                 |                 | (Packet #6)
  |                 |---------------->|                 | INFORMATIONAL response (HDR, SK {})
  |                 |                 |                 | (Judgment #6)
  |                 |                 |                 |
  |<----------------+=================+-----------------| IPsec {Echo Request}
  |                 |                 |                 | (Packet #7) (Judgment #7)
  |-----------------+=================+---------------->| IPsec {Echo Reply}
  |                 |                 |                 | (Packet #8) (Judgment #8)
  |                 |                 |                 |
  V                 V                 V                 V
```

Packet #1	See Common Packet #1

Packet #2	See Common Packet #5
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See Common Packet #11
Packet #6	See below
Packet #7	See Common Packet #21
Packet #8	See Common Packet #25

- Packet #6: INFORMATIONAL request

IPv6 Header	Same as the Common Packet #17	
UDP Header	Same as the Common Packet #17	
IKEv2 Header	Same as the Common Packet #17	
E Payload	Other fields are same as the Common Packet #17	
	Next Payload	42 (Delete)
Delete Payload	Next Payload	0 (last)
	Critical	0
	Reserved	0
	Payload Length	16
	Procotol ID	1 (IKE_SA)
	SPI Size	0
	# of SPIs	0
	Security Parameter Index(es) (SPI)	empty

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TH2 transmits an Echo Request to TH1.
6. Observe the messages transmitted on Link B.
7. TH1 transmits an Echo Reply to TH2.
8. Observe the messages transmitted on Link A.
9. TN1 transmits a CREATE_CHILD_SA request to rekey IKE_SA. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value.
10. Observe the messages transmitted on Link A.
11. TN1 transmits an INFORMATIONAL request with a Delete payload which has 1 (IKE_SA) in the Protocol ID field, zero in the SPI Size field and zero in the # of SPIs field.
12. Observe the messages transmitted on Link A.
13. TH2 transmits an Echo Request to TH1. TN1 forwards an Echo Request with IPsec ESP with corresponding algorithms inherited from the replaced IKE_SA.
14. Observe the messages transmitted on Link B.
15. TH1 transmits an Echo Reply to TH2.
16. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms. And the proposal in the SA payload includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA Responder’s SPI value in the SPI field.

Step 12: Judgment #6
The NUT responds with an INFORMATIONAL response with no payloads.

Step 14: Judgment #3
The NUT forwards an Echo Request.

Step 16: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms inherited from the replaced IKE_SA.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.5: Receiving Multiple Transform

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple transform to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Packet #2)
  |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH SAr2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                 | (Packet #3)
  |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr })
  |                 | (Judgment #3)
  V                 V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

From part A to part D, TN1 transmits an IKE_SA_INIT request including a SA payload which contains the transforms as follows:

	IKE_SA_INIT exchanges Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	**ENCR_AES_CBC** **ENCR_3DES**	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	ENCR_3DES	**PRF_AES128_CBC** **PRF_HMAC_SHA1**	AUTH_HMAC_SHA1_96	Group 2
Part C	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96** **AUTH_HMAC_SHA1_96**	Group 2

- Packet #3 CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #11	
UDP Header	Same as the Common Packet #11	
IKEv2 Header	Same as the Common Packet #11	
SA Payload	Other fields are same as the common packet #11	
	SA Proposals	See SA Table below
Ni, Nr Payload	Same as the Common Packet #11	

Proposal #1	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	According to above configuration
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo Random Function (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: MultipleIntegrity Algorithm (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Multiple D-H Group (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part D

Step 20: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.6: Receiving Multiple Proposal

Purpose:

To verify an IKEv2 device properly handles a CREATE_CHILD_SA request with multiple proposal to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.7, 2.8 and 3.3

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT             TN1
 (SGW)           (SGW)
   |               |
   |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |               | (Packet #1)
   |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |               | (Judgment #1)
   |               |
   |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |               | (Packet #2)
   |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |               | (Judgment #2)
   |               |
   |<--------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
   |               | (Packet #3)
   |-------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
   |               | (Judgment #3)
   V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See below

TN1 transmits a CREATE_CHILD_SA request including a SA payload which contains the two proposals as follows:

	IKE_SA_INIT exchanges Algorithms					
	Proposals	Protocol ID	Encryption	PRF	Integrity	D-H Group
Part A	Proposal #1	IKE	ENCR_AES_CBC	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part B	Proposal #1	IKE	ENCR_3DES	PRF_AES128_CBC	AUTH_HMAC_SHA1_96	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

Part C	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	**AUTH_AES_XCBC_96**	Group 2
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2
Part D	Proposal #1	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	**Group 14**
	Proposal #2	IKE	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	Group 2

- Packet #3: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #11	
UDP Header	Same as the Common Packet #11	
IKEv2 Header	Same as the Common Packet #11	
SA Payload	Other fields are same as the common packet #11	
	SA Proposals	See SA Table below
Ni, Nr Payload	Same as the Common Packet #11	

Proposal #1	SA Proposal	Next Payload		2 (more)
		Reserved		0
		Proposal Length		44
		Proposal #		1
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	According to above configuration
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	According to above configuration
Proposal #2	SA Proposal	Next Payload		0 (last)
		Reserved		0
		Proposal Length		44
		Proposal #		2
		Protocol ID		1 (IKE)
		SPI Size		0
		# of Transforms		5
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	1 (ENCR)
			Reserved	0
			Transform ID	3 (3DES)
		SA Transform	Next Payload	3 (more)
			Reserved	0

			Transform Length	8
			Transform Type	2 (PRF)
			Reserved	0
			Transform ID	2 (HMAC_SHA1)
		SA Transform	Next Payload	3 (more)
			Reserved	0
			Transform Length	8
			Transform Type	3 (INTEG)
			Reserved	0
			Transform ID	2 (HMAC_SHA1_96)
		SA Transform	Next Payload	0 (last)
			Reserved	0
			Transform Length	8
			Transform Type	4 (D-H)
			Reserved	0
			Transform ID	2 (1024 MODP Group)

Part A: Multiple Encryption Algorithms (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
6. Observe the messages transmitted on Link A.

Part B: Multiple Pseudo Random Function (BASIC)

7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A.
9. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A.
11. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
12. Observe the messages transmitted on Link A.

Part C: Multiple Integrity Algorithms (BASIC)

13. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
14. Observe the messages transmitted on Link A.
15. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
16. Observe the messages transmitted on Link A.
17. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.
18. Observe the messages transmitted on Link A.

Part D: Multiple D-H Group (BASIC)

19. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
20. Observe the messages transmitted on Link A.
21. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
22. Observe the messages transmitted on Link A.
23. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT.

24. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part B

Step 8: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 10: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 12: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part C

Step 14: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 18: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Part D

Step 20: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 22: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 24: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.7: Changing RPFs when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly handles CREATE_CHILD_SA to rekey IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
  Configure the devices according to the Common Configuration except for *Italic* parameters.

	IKE_SA Rekeying Algorithms			
	Encryption	**PRF**	**Integrity**	**D-H Group**
Part A	ENCR_3DES	PRF_HMAC_SHA1	AUTH_HMAC_SHA1_96	***Group 14***

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                | (Packet #3)
  |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr})
  |                | (Judgment #3)
  |                |
  |<---------------| INFORMATION Request (HDR, SK {})
  |                | (Packet #4)
  |--------------->| INFORMATIONAL response (HDR, SK {})
  |                | (Judgment #4)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Packet #3	See Common Packet #11
Packet #4	See Common Packet #17 (encrypted by the new IKE_SA)

Packet #3: CREATE_CHILD_SA request

Packet #3 is same as Common Packet #11 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0
	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	14 (2048 MODP Group)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value.
6. Observe the messages transmitted on Link A.
7. TN1 transmits an INFORMATIONAL request with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.
8. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3

The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 14” as proposed algorithms. And the proposal in the SA payload includes 1 (IKE) in the Protocol ID field, 8 in the SPI size field and rekeyed IKE_SA Responder's SPI value in the SPI field.

Step 8: Judgment #4

The NUT responds with an INFORMATIONAL response with no payloads protected by the new IKE_SA and the Message ID field in the IKE header is zero.

Possible Problems:

- none

Test IKEv2.SGW.R.1.2.6.8: D-H transform NONE when rekeying the IKE_SA

Purpose:

To verify an IKEv2 device properly handles D-H transform NONE when rekeying IKE_SA.

References:

- [RFC 4306] - Sections 2.8
- [RFC 4718] - Sections 5.12

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<---------------| CREATE_CHILD_SA request (HDR, SK {SA, Ni})
  |                | (Packet #3)
  |-------X        | CREATE_CHILD_SA response (HDR, SK {N(NO_PROPOSAL_CHOSEN})
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #11

Packet #3: CREATE_CHILD_SA request

Packet #3 is same as Common Packet #11 except SA Transform proposed in each test.

Part A:

SA Transform of Tranform Type D-H is replaced by the following SA Transfrom.

SA Transform	Next Payload	0 (last)
	Reserved	0

	Transform Length	8
	Transform Type	4 (D-H)
	Reserved	0
	Transform ID	0 (NONE)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits a CREATE_CHILD_SA request including a SA payload. A proposal in the SA payload contains 1 (IKE) in the Protocol ID field, 8 in the SPI size field and the rekeyed IKE_SA Initiator's SPI value. The message proposes D-H transform NONE.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including a Notify payload of type NO_PROPOSAL_CHOSEN.

Possible Problems:

- none

Group 2.7. Creating New CHILD_SA with the CREATE_CHILD_SA Exchange

Test IKEv2.SGW.R.1.2.7.1: Receipt of cryptographically protected message on the new SA

Purpose:

To verify an IKEv2 device properly recognizes the lifetime of CHILD_SAs.

References:

- [RFC 4306] - Sections 2.8

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1           NUT           TN1           TH2           TH3
(Host)        (SGW)         (SGW)         (Host)        (Host)
  |             |             |             |             |
  |             |<------------|             |             | IKE_SA_INIT request (HDR, SAi1,
  |             |             |             |             |                          KEi, Ni)
  |             |             |             |             | (Packet #1)
  |             |------------>|             |             | IKE_SA_INIT Response (HDR, SAr1,
  |             |             |             |             |                           KEr, Nr)
  |             |             |             |             | (Judgment #1)
  |             |             |             |             |
  |             |<------------|             |             | IKE_AUTH request (HDR, SK {IDi, AUTH,
  |             |             |             |             |                  SAi2, TSi, TSr})
  |             |             |             |             | (Packet #2)
  |             |------------>|             |             | IKE_AUTH Response (HDR, SK {IDr, AUTH,
  |             |             |             |             |                   SAr2, TSi, TSr})
  |             |             |             |             | (Judgment #2)
  |             |             |             |             |
  |<------------+=============+-------------|             | IPsec {Echo Request}
  |             |             |             |             | (Packet #3) (Judgment #3)
  |-------------+=============+------------>|             | IPsec {Echo Reply}
  |             |             |             |             | (Packet #4) (Judgment #4)
  |             |             |             |             |
  |       X-----+=============+---------------------------| IPsec {Echo Request}
  |             |             |             |             | (Packet #5) (Judgment #5)
  |-------------+======X      |             |             | IPsec {Echo Request}
  |             |             |             |             | (Packet #6) (Judgment #6)
  |             |             |             |             |
  |             |<------------|             |             | CREATE_CHILD_SA request (HDR,
  |             |             |             |             |                SK{SA, Ni, TSi, TSr})
  |             |             |             |             | (Packet #7)
  |             |------------>|             |             | CREATE_CHILD_SA response (HDR,
  |             |             |             |             |                SK{SA, Nr, TSi, TSr})
  |             |             |             |             | (Judgment #7)
  |             |             |             |             |
  |<------------+=============+-------------|             | IPsec {Echo Request}
  |             |             |             |             | (Packet #8) (Judgment #8)
  |-------------+=============+------------>|             | IPsec {Echo Reply}
  |             |             |             |             | (Packet #9) (Judgment #9)
  |             |             |             |             |
  |<------------+=============+---------------------------| IPsec {Echo Request}
  |             |             |             |             | (Packet #10) (Judgment #10)
  |-------------+=============+-------------------------->| IPsec {Echo Reply}
  |             |             |             |             | (Packet #11) (Judgment #11)
  |             |             |             |             |
  V             V             V             V             V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See Common Packet #21
Packet #4	See Common Packet #25
Packet #5	See below
Packet #6	See below
Packet #7	See below
Packet #8	See Common Packet #21
Packet #9	See Common Packet #25
Packet #10	See below
Packet #11	See below

- Packet #2: IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Same as the Common Packet #5	
N Payload	Same as the Common Packet #5	
SA Payload	Same as the Common Packet #5	
TSi Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #5	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH2's Global Address on Link B
		Ending Address	TH2's Global Address on Link B

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH1's Global Address on Link Y
		Ending Address	TH1's Global Address on Link Y

- Packet #5: Echo Request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #6: Echo Request

IPv6 Header	Source Address	TH1's Global Address
	Distination Address	TH3's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #7: CREATE_CHILD_SA request

IPv6 Header	Same as the Common Packet #4	
UDP Header	Same as the Common Packet #4	
IKEv2 Header	Same as the Common Packet #4	
E Payload	Same as the Common Packet #4	
IDi Payload	Same as the Common Packet #4	
AUTH Payload	Same as the Common Packet #4	
N Payload	Same as the Common Packet #4	
SA Payload	Same as the Common Packet #4	
TSi Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below
TSr Payload	Other fields are same as the Common Packet #4	
	Traffic Selectors	See below

TSi Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH3's Global Address on Link B
		Ending Address	TH3's Global Address on Link B

TSr Payload	Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
		IP Protocol ID	0 (any)
		Selector Length	40
		Start Port	0
		End Port	65535
		Starting Address	TH1's Global Address on Link Y
		Ending Address	TH1's Global Address on Link Y

- Packet #10: Echo Request

IPv6 Header	Source Address	TN1's Global Address on Link X
	Destination Address	NUT's Global Address on Link A
ESP	Security Parameter Index	CHILD_SA's SPI value used by this message
	Sequence Number	The value incremented the previous encrypted packet's Sequence Number by one.
	Payload Data	Subsequent data encrypted by underlying encryption algorithm
	Padding	Any value which to be a multiple of the encryption block size
	Pad Length	The length of the Padding field
	Next Header	41 (IPv6)
	Integrity Check Value	The checksum must be valid by calculation according to the manner described in RFC.
IPv6 Header	Source Address	TH3's Global Address
	Destination Address	TH1's Global Address
ICMPv6 Header	Type	128
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

- Packet #11: Echo Reply

IPv6 Header	Source Address	TH1's Global Address
	Distination Address	TH3's Global Address
ICMPv6 Header	Type	129
	Code	0
	Identifier	any
	Sequence Number	any
	Payload Data	0x0000000000000000

Part A: (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link B.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link B.
5. TH2 transmits an Echo Request packet to TH1.
6. Observe the messages transmitted on Link A.
7. TH1 transmits an Echo Reply packet to TH2.
8. Observe the messages transmitted on Link B.
9. TH3 transmits an Echo Request packet to TH1.
10. Observe the messages transmitted on Link A.
11. TH1 transmits an Echo Request packet to TH3.
12. Observe the messages transmitted on Link B.
13. TN1 starts to negotiate new CHILD_SA with the NUT by sending CREATE_CHILD_SA request.
14. Observe the messages transmitted on Link B.
15. TH2 transmits an Echo Request packet to TH1.
16. Observe the messages transmitted on Link A.
17. TH1 transmits an Echo Reply packet to TH2.
18. Observe the messages transmitted on Link B.
19. TH3 transmits an Echo Request packet to TH1.
20. Observe the messages transmitted on Link A.
21. TH1 transmits an Echo Reply packet to TH3.
22. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Step 10: Judgment #5
The NUT never forwards an Echo Request.

Step 12: Judgment #6
The NUT never forwards an Echo Reequest with IPsec ESP using the first negotiated algorithms.

Step 14: Judgment #7
The NUT transmits a CREATE_CHILD_SA response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 16: Judgment #8
The NUT forwards an Echo Request.

Step 18: Judgment #9
The NUT forwards an Echo Reply with IPsec ESP using the first negotiated algorithms.

Step 20: Judgment #10
The NUT forwards an Echo Request.

Step 22: Judgment #11
The NUT forwards an Echo Reply with IPsec ESP using the second negotiated algorithms.

Possible Problems:

- None

Group 2.8. Error Handling

Test IKEv2.SGW.R.1.2.8.1: AUTHENTICATION_FAILED

Purpose:

To verify an IKEv2 device properly handles AUTHENTICATION_FAILED message.

References:

- [RFC 4306] - Sections 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |------->        | IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                |   or
  |--------------->| IKE_AUTH Response (HDR, N(AUTHENTICATION_FAILED))
  |                | (Judgment #2)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2 (Part A): IKE_AUTH request

IPv6 Header	Same as the Common Packet #5	
UDP Header	Same as the Common Packet #5	
IKEv2 Header	Same as the Common Packet #5	
E Payload	Same as the Common Packet #5	
IDi Payload	Same as the Common Packet #5	
AUTH Payload	Other fields are same as the Common Packet #5	
	Payload Length	8
	Auth Method	2 (SK_MIC)
	Authentication Data	empty

N Payload	Same as the Common Packet #5
SA Payload	Same as the Common Packet #5
TSi Payload	Same as the Common Packet #5
TSr Payload	Same as the Common Packet #5

- Packet #2 (Part B): IKE_AUTH request

IPv6 Header		Same as the Common Packet #5
UDP Header		Same as the Common Packet #5
IKEv2 Header		Same as the Common Packet #5
E Payload		Same as the Common Packet #5
IDi Payload		Same as the Common Packet #5
AUTH Payload		Other fields are same as the Common Packet #5
	Payload Length	28
	Auth Method	1 (RSA_DS)
	Authentication Data	Same data as the common packet #5 (caluculated by using SK_MIC)
N Payload		Same as the Common Packet #5
SA Payload		Same as the Common Packet #5
TSi Payload		Same as the Common Packet #5
TSr Payload		Same as the Common Packet #5

Part A Invalid Authentication Data (ADVANCED)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request which has an invalid Authentication Data in AUTH payload to the NUT.
4. Observe the messages transmitted on Link A..

Part B Invalid Auth method (ADVANCED)

5. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
6. Observe the messages transmitted on Link A..
7. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request which has an invalid Auth Method in AUTH payload to the NUT.
8. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT does not transmit an IKE_AUTH response or transmits an IKE_AUTH response with Notify payload of type AUTHENTICATION_FAILED without encryption to the TN1.

Part B

Step 6: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 8: Judgment #2

The NUT does not transmit an IKE_AUTH response or transmits an IKE_AUTH response with Notify payload of type AUTHENTICATION_FAILED without encryption to the TN1.

Possible Problems:

- None.

Group 2.9. Non zero RESERVED fields

Test IKEv2.SGW.R.1.2.9.1: Non zero RESERVED fields in CREATE_CHILD_SA request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
   NUT              TN1
  (SGW)            (SGW)
    |                |
    |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
    |                | (Packet #1)
    |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
    |                | (Judgment #1)
    |                |
    |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
    |                | (Packet #2)
    |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
    |                | (Judgment #2)
    |                |
    |<---------------| CREATE_CHILD_SA request (HDR, SK {N, SA, Ni, TSi, TSr})
    |                | (Packet #3)
    |--------------->| CREATE_CHILD_SA response (HDR, SK {SA, Nr, TSi, TSr})
    |                | (Judgment #3)
    V                V

N: REKEY_SA
N+: USE_TRANSPORT_MODE
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #5	See Common Packet #15 All RESERVED fields are set to one.

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits a CREATE_CHILD_SA request including a Notify Payload of type REKEY_SA and rekeyed CHILD_SA's SPI value in the SPI field to the NUT. All RESERVED fields are set to one.
6. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as accepted algorithms.

Step 6: Judgment #3
The NUT transmits a CREATE_CHILD_SA response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- None.

Group 3. The INFORMATIONAL Exchange

Group 3.1. Header and Payload Formats

Test IKEv2.SGW.R.1.3.1.1: Sending INFORMATIONAL response

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.1.2 and 1.4

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT              TN1
(SGW)            (SGW)
  |                |
  |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                | (Packet #1)
  |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                | (Judgment #1)
  |                |
  |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                | (Packet #2)
  |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                | (Judgment #2)
  |                |
  |<---------------| INFORMATION request (HDR, SK { })
  |                | (Packet #3)
  |--------------->| INFORMATIONAL response (HDR, SK { })
  |                | (Judgment #3)
  |                |
  V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #17

Part A: IKE Header Format (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A..
3. After reception of IKE_SA_INIT_SA response from the NUT, TN1 transmits an

IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A..
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
6. Observe the messages transmitted on Link A..

Part B: Encrypted Payload Format (BASIC)
7. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
8. Observe the messages transmitted on Link A..
9. After reception of IKE_SA_INIT_SA response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
10. Observe the messages transmitted on Link A..
11. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads to the NUT.
12. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 7: Judgment #3
The NUT transmits an INFORMATIONAL response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 177 Header format

- An IKE_SA Initiator’s SPI field set to same as the IKE_SA_INIT request’s IKE_SA Initiator’s SPI field value.
- An IKE_SA Responder’s SPI field set to same as the IKE_SA_INIT response’s IKE_SA Responder’s SPI field value.
- A Next Payload field set to Encrypted Payload (46).

- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to INFORMATIONAL (37).
- A Flags field set to (00000100)2 = (4)10.
- A Message ID field set to the same value as corresponding IKEv2 request message's Message ID.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 9: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 11: Judgment #2

The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 14: Judgment #3

The NUT transmits an INFORMATIONAL response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 178 Encrypted payload

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire

message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Possible Problems:

- None.

Group 3.2. Use of Retransmission Timers

Test IKEv2.SGW.R.1.3.2.1: Receipt of retransmitted INFORMATIONAL request

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.1.2, 1.4 and 2.1

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |               | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |               | (Judgment #1)
  |               |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |               | (Packet #2)
  |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |               | (Judgment #2)
  |               |
  |<--------------| INFORMATIONAL request (HDR, SK { })
  |               | (Packet #3)
  |-------------->| INFORMATIONAL response (HDR, SK { })
  |               | (Judgment #3)
  |               |
  |               * wait until retrans timer expires
  |-------------->| INFORMATIONAL response (HDR, SK { })
  |               | (Judgment #4)
  |               |
  |<--------------| INFORMATIONAL request (HDR, SK { })
  |               | (Packet #4)
  |-------------->| INFORMATIONAL response (HDR, SK { })
  |               | (Judgment #5)
  |               |
  V               V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #17

Packet #4	See Common Packet #17 (same Message ID as packet #3)

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_SA_INIT response from the NUT, TN1 transmits an IKE_AUTH request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an INFORMATIONAL request with no payloads.
6. Observe the messages transmitted on Link A.
7. Observe the messages transmitted on Link A.
8. TN1 transmits an INFORMATIONAL request with no payloads. The Message ID is the same as Step 5.
9. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Step 7: Judgment #4
The NUT transmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Step 9: Judgment #5
The NUT transmits an INFOMATIONAL response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Group 3.3. Non zero RESERVED fields

Test IKEv2.SGW.R.1.3.3.1: Non RESERVED fields in INFORMATIONAL request

Purpose:

To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.In addition, set IKE_SA Lifetime to 300 seconds and set CHILD_SA Lifetime to 30 seconds.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT               TN1
(SGW)             (SGW)
  |                 |
  |<--------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                 | (Packet #1)
  |-------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                 | (Judgment #1)
  |                 |
  |<--------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                 | (Packet #2)
  |-------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                 | (Judgment #2)
  |                 |
  |<--------------| INFORMATIONAL request (HDR, SK {})
  |                 | (Packet #3)
  |-------------->| INFORMATIONAL response (HDR, SK {})
  |                 | (Judgment #3)
  |                 |
  V                 V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5
Packet #3	See Common Packet #17 All RESERVED fields are set to one.

Part A: (BASIC)

1. TN1 starts to negotiate with NUT by sending IKE_SA_INIT request.
2. Observe the messages transmitted on Link A.
3. After reception of IKE_AUTH response from the NUT, TN1 transmits an IKE_AUTH

request to the NUT.
4. Observe the messages transmitted on Link A.
5. After reception of IKE_AUTH response from the NUT, TN1 transmits an INFORMATIONAL request with no payloads. All RESERVED fields in the message are set to one.
6. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as accepted algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as accepted algorithms.

Step 6: Judgment #3
The NUT transmits an INFOMATIONAL Response followed by an Encrypted payload with no payloads contained in it.

Possible Problems:

- None

Section 2.2.2. Endpoint to Security Gateway Tunnel
Group 1. The Initial Exchanges

Group 1.1. Header and Payload Formats

Test IKEv2.SGW.R.2.1.1.1: Sending IKE_AUTH response

Purpose:

To verify an IKEv2 device transmits IKE_AUTH request using properly Header and Payloads format

References:

- [RFC 4306] - Sections 1.2, 2.15, 3.1, 3.2, 3.3, 3.5, 3.8, 3.10, 3.13 and 3.14

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
  NUT              TN1
 (SGW)          (End-Node)
   |                |
   |<---------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
   |                | (Packet #1)
   |--------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
   |                | (Judgment #1)
   |                |
   |<---------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
   |                | (Packet #2)
   |--------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
   |                | (Judgment #2)
   |                |
   V                V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5

Part A: IKE Header Format (BASIC)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A..
3. TN1 transmits an IKE_SA_INIT request to NUT.
4. Observe the messages transmitted on Link A..

Part B: Encrypted Payload Format (BASIC)

5. TN1 transmits an IKE_SA_INIT request to NUT.
6. Observe the messages transmitted on Link A..
7. TN1 transmits an IKE_SA_INIT request to NUT.

8. Observe the messages transmitted on Link A..

Part C: IDr Payload Format (BASIC)

9. TN1 transmits an IKE_SA_INIT request to NUT.
10. Observe the messages transmitted on Link A..
11. TN1 transmits an IKE_SA_INIT request to NUT.
12. Observe the messages transmitted on Link A..

Part D: AUTH Payload Format (BASIC)

13. TN1 transmits an IKE_SA_INIT request to NUT.
14. Observe the messages transmitted on Link A..
15. TN1 transmits an IKE_SA_INIT request to NUT.
16. Observe the messages transmitted on Link A..

Part E: SA Payload Format (BASIC)

17. TN1 transmits an IKE_SA_INIT request to NUT.
18. Observe the messages transmitted on Link A..
19. TN1 transmits an IKE_SA_INIT request to NUT.
20. Observe the messages transmitted on Link A..

Part F: TSi Payload Format (BASIC)

21. TN1 transmits an IKE_SA_INIT request to NUT.
22. Observe the messages transmitted on Link A..
23. TN1 transmits an IKE_SA_INIT request to NUT.
24. Observe the messages transmitted on Link A..

Part G: TSr Payload Format (BASIC)

25. TN1 transmits an IKE_SA_INIT request to NUT.
26. Observe the messages transmitted on Link A..
27. TN1 transmits an IKE_SA_INIT request to NUT.
28. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted IKE Header containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Initiator's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       IKE_SA Responder's SPI                  !
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Next Payload ! MjVer ! MnVer ! Exchange Type !     Flags     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                          Message ID                           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                            Length                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 179 Header format

- An IKE_SA Initiator's SPI field set to same as the IKE_SA_INIT request's IKE_SA Initiator's SPI field value.
- An IKE_SA Responder's SPI field set to same as the IKE_SA_INIT response's IKE_SA Responder's SPI field value.
- A Next Payload field set to Encrypted Payload (46).
- A Major Version field set to 2.
- A Minor Version field set to zero.
- An Exchange Type field set to IKE_AUTH (35).
- A Flags field set to (00010000)2 = (16)10.
- A Message ID field set to 1.
- A Length field set to the length of the message (header + payloads) in octets.

Part B

Step 6: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 8: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted Encrypted Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                     Initialization Vector                     !
!         (length is block size for encryption algorithm)       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!               !             Padding (0-255 octets)            !
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
!                                               !  Pad Length   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 180 Encrypted payload

- A Next Payload field set to IDr Payload (36).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length in octets of the header, IV, Encrypted IKE Payloads, Padding, Pad Length, and Integrity Check sum Data.
- An Initialization Vector field set to a randomly chosen value whose length is equal to the block length of the underlying encryption algorithm. It is 64 bits length in ENCR_3DES case.
- An Encrypted IKE Payloads field set to subsequent payloads encrypted by ENCR_3DES.
- A Padding field set to any value which to be a multiple of the encryption block size. It is 64 bits length in ENCR_3DES case.
- A Pad Length field set to the length of the Padding field.
- An Integrity Checksum Data set to the cryptographic checksum of the entire message. It is 96 bits length in AUTH_HMAC_SHA1_96 case. The checksum must be valid by calculation according to the manner described in RFC.

Part C

Step 10: Judgment #1

The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 12: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted ID Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ID Type     !                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Identification Data                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 181 ID Payload format

- A Next Payload field set to AUTH Payload (39).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 24 bytes for ID_IPV6_ADDR.
- An ID Type field set to ID_IPV6_ADDR (5).
- A RESERVED field set to zero.
- An Identification Data field set to the NUT address.

Part D

Step 14: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 16: Judgment #2
The NUT transmits an IKE_AUTH response including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Auth Method   !                RESERVED                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Authentication Data                      ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 182 AUTH Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload. It is 28 bytes for PRF_HMAC_SHA1.
- An Auth Method field set to Shared Key Message Integrity Code (2).
- A RESERVED field set to zero.
- An Authentication Data field set to correct authentication value according to the manner described in RFC. It is 160 bytes length in PRF_HMAC_SHA1 case.

Part E

Step 18: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 20: Judgment #2

```
                         1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
    ! Next    44   !0!     0       ! Length          40             !               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---           |
    !       0       !       0       ! Length          36             ! |            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |            |
    ! Number    1   ! Prot ID   3   ! SPI Size  4   ! Trans Cnt  3   ! |            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |            |
    ! SPI value                                                     ! |            |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |            |
  | !       3       !       0       ! Length          8             ! |            |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |    |SA Payload
  | ! Type  1  (EN) !       0       ! Transform ID    3     (3DES) ! | Proposal |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |            |
  | !       3       !       0       ! Length          8             ! |            |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |     |
  | ! Type  3  (IN) !       0       ! Transform ID    2     (SHA1) ! |            |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |            |
  | !       0       !       0       ! Length          8             ! |            |
Transform | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |     |
  | ! Type  5  (ESN)!       0       ! Transform ID    0      (No)  ! |            |
--- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ --------------
```

Figure 183 SA Payload contents

The NUT transmits an IKE_AUTH response including properly formatted SA Payload containing following values (refer following figures):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                          <Proposals>                          ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 184 SA Payload format

- A Next Payload field set to TSi Payload (44).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.

A Proposals field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 2 !   RESERVED    !         Proposal Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Proposal #    !  Protocol ID  !    SPI Size   !# of Transforms!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                        <Transforms>                           ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 185 Proposal sub-structure format

Proposal #1

- A 0 or 2 field set to zero (last).
- A RESREVD field set to zero.
- A Proposal Length field set to length of this proposal, including all transforms and attributes. It is 36 bytes according to Common Configuration.
- A Proposal # field set to 1.
- A Protocol ID field set to ESP (3).
- A SPI Size field set to 4.
- A # of Transforms field set to 3.
- A SPI field set to the sending entity's SPI (4 octets value)

Transform field set to following (There are 3 Transform Structures).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! 0 (last) or 3 !   RESERVED    !        Transform Length       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Transform Type !   RESERVED    !          Transform ID         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                      Transform Attributes                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 186 Transform sub-structure format

Transform #1

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ENCR_3DES.
- A Transform Type field set to ENCR (1).
- A RESERVED field set to zero.
- A Transform ID set to ENCR_3DES (3).

Transform #2

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including

Header and Attribute. It is 8 bytes for AUTH_HMAC_SHA1.

- A Transform Type field set to INTEG (3).
- A RESERVED field set to zero.
- A Transform ID set to AUTH_HMAC_SHA1 (2).

Transform #3

- A 0 or 3 field set to zero if this structure is the last transform, otherwise set to 3.
- A RESERVED field set to zero.
- A Transform Length set to length of the Transform Substructure including Header and Attribute. It is 8 bytes for ESN.
- A Transform Type field set to ESN (5).
- A RESERVED field set to zero.
- A Transform ID set to No Extended Sequence Numbers (0).

Part F

Step 22: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 24: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSi Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 187 TSi Payload format

- A Next Payload field set to TSr Payload (45).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Starting Address*                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                         Ending Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 188 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to TN1 address.
- A Ending Address field set to greater thatn or equal to TN1 address.

Part G

Step 26: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 28: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted TSr Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C!  RESERVED   !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Number of TSs !                 RESERVED                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       <Traffic Selectors>                     ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 189 TSr Payload format

- A Next Payload field set to zero.
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A Number of TSs field set to 1.
- A RESERVED field set to zero.

Traffic Selectors field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   TS Type     !IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Starting Address*                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Ending Address*                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 190 Traffic Selector

- A TS Type set to TS_IPV6_ADDR_RANGE (8).
- An IP Protocol ID field set to zero.
- A Selector Length field set to length of this Traffic Selector Substructure including the header. It is 40 bytes for TS_IPV6_ADDR_RANGE.
- A Start Port field set to zero.
- An End Port field set to 65535.
- A Starting Address field set to less than or equal to Prefix B.
- An Ending Address field set to less than or equal to Prefix B.

Possible Problems:

- IKE_AUTH response has following packet format.It may have additional payloads described below. Additional payloads can be ignored by this test. The order of payload may be different from this sample.

```
IDr, [CERT+],
AUTH,
[CP(CFG_REPLY)],
[N(IPCOMP_SUPPORTED)],
[N(USE_TRANSPORT_MODE)],
[N(ESP_TFC_PADDING_NOT_SUPPORTED)],
[N(NON_FIRST_FRAGMENTS_ALSO)],
SA, TSi, TSr,
[N(ADDITIONAL_TS_POSSIBLE)],
[V+]
```

- Each of transforms can be located in the any order.

Test IKEv2.SGW.R.2.1.1.2: Use of CHILD_SA

Purpose:

To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 1.2

Test Setup:

- Network Topology
  Connect the devices according to the Common Topology.
- Configuration
  In each part, configure the devices according to the Common Configuration.
- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1
(Host)             (SGW)           (End-Node)
  |                  |                  |
  |                  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  | (Packet #1)
  |                  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  | (Judgment #1)
  |                  |                  |
  |                  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                  |                  | (Packet #2)
  |                  |----------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, SAr2, TSi, TSr})
  |                  |                  | (Judgment #2)
  |                  |                  |
  |<-----------------+==================| IPsec {Echo Request}
  |                  |                  | (Judgment #3)
  |------------------+=================>| IPsec {Echo Reply}
  |                  |                  | (Judgment #4)
  |                  |                  |
  V                  V                  V
```

Packet #1	See Common Packets
Packet #2	See Common Packets

Part A (BASIC)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A..
3. TN1 transmits an IKE_SA_INIT response to NUT.
4. Observe the messages transmitted on Link A..
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to TH1.
6. Observe the messages transmitted on Link A..
7. TH1 transmits an Echo Reply to TN1.
8. Observe the messages transmitted on Link B...

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3

The NUT forwards an Echo Request.

Step 8 Judgment #4

The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- None.

Group 1.2. Requesting an Internal Address on a Remote Network

Test IKEv2.SGW.R.2.1.2.1: Receipt of CFG_REQUEST

Purpose:

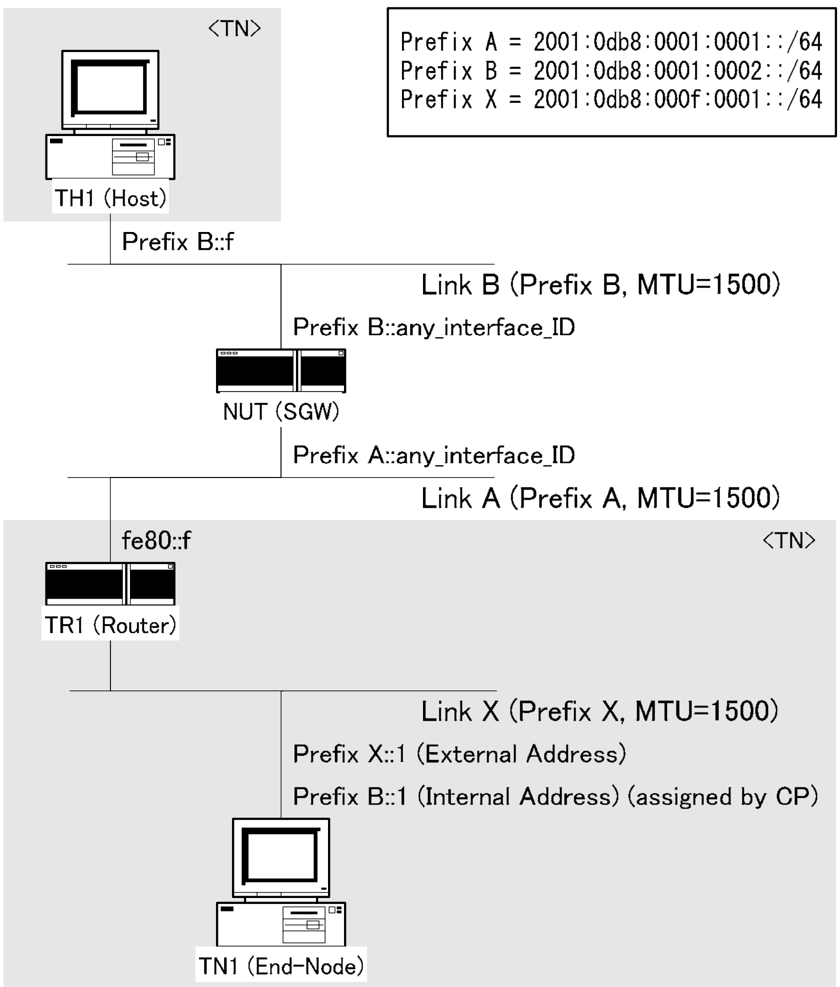
To verify an IKEv2 device transmits IKE_AUTH request using properly eader and Configuration Payload format

References:

- [RFC 4306] - Sections 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic

selector. Configure NUT to transmit CFG_REPLY for INTERNAL_IP6_ADDRESS. Its IPv6 address is Prefix B::1/128. The traffic selector must be configured by the following table. NUT must narrow Traffic Selector to the following address range.

	Traffic Selector					
	Source			**Destination**		
	Address Range	**Next Layer Protocol**	**Port Range**	**Address Range**	**Next Layer Protocol**	**Port Range**
Inbound	TN1 (internal address)	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1 (internal address)	ANY	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                TN1
(SGW)            (End-Node)
  |                  |
  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  | (Packet #1)
  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  | (Judgment #1)
  |                  |
  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH, CP(CFG_REQUEST), SAi2, TSi, TSr})
  |                  | (Packet #2)
  |----------------->| IKE_AUTH Response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
  |                  | (Judgment #2)
  |                  |
  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request packet

IPv6 Header	Same as Common Packet #5	
UDP Header	Same as Common Packet #5	
IKEv2 Header	Same as Common Packet #5	
E Payload	Same as Common Packet #5	
IDi Payload	Same as Common Packet #5	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #5	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	12
	CFG Type	1 (CFG_REQUEST)
	RESERVED	0
	Configuration Attributes	See below
SA Payload	Same as Common Packet #5	
TSi Payload	Other fields are same as Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #5	

Configuration Attributes	Reserved	0
	Attribute Type	INTERNAL_IP6_ADDRESS
	Length	0

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	::
	Ending Address	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Part A: (ADVANCED)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A..
3. TN1 transmits an IKE_SA_INIT request to NUT.
4. Observe the messages transmitted on Link A..

Observable Results:

Part A

Step 2: Judgment #1

The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2

The NUT transmits an IKE_AUTH response including properly formatted AUTH Payload containing following values:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
! Next Payload  !C! RESERVED    !         Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   CFG Type    !                    RESERVED                   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                   Configuration Attributes                    ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 191 Configuration Payload format

- A Next Payload field set to SA Payload (33).
- A Critical field set to zero.
- A RESERVED field set to zero.
- A Payload Length field set to length of the current payload.
- A CFG Type field set to CFG_REPLY (2).
- A RESERVED field set to zero.

A Configuration Attributes field set to following.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!R|         Attribute Type      !            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                             Value                             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 192 Configuration Attributes format

Configuration Attribute #1

- Reserved field is set to zero.
- Attribute Type field is set to INTERNAL_IP6_ADDRESS (8).
- Length field is set to 17.
- Value field is set to Prefix B::1 as IPv6 address and 128 as prefix-length.

Possible Problems:

- None.

Test IKEv2.SGW.R.2.1.2.2: Use of CHILD_SA

Purpose:

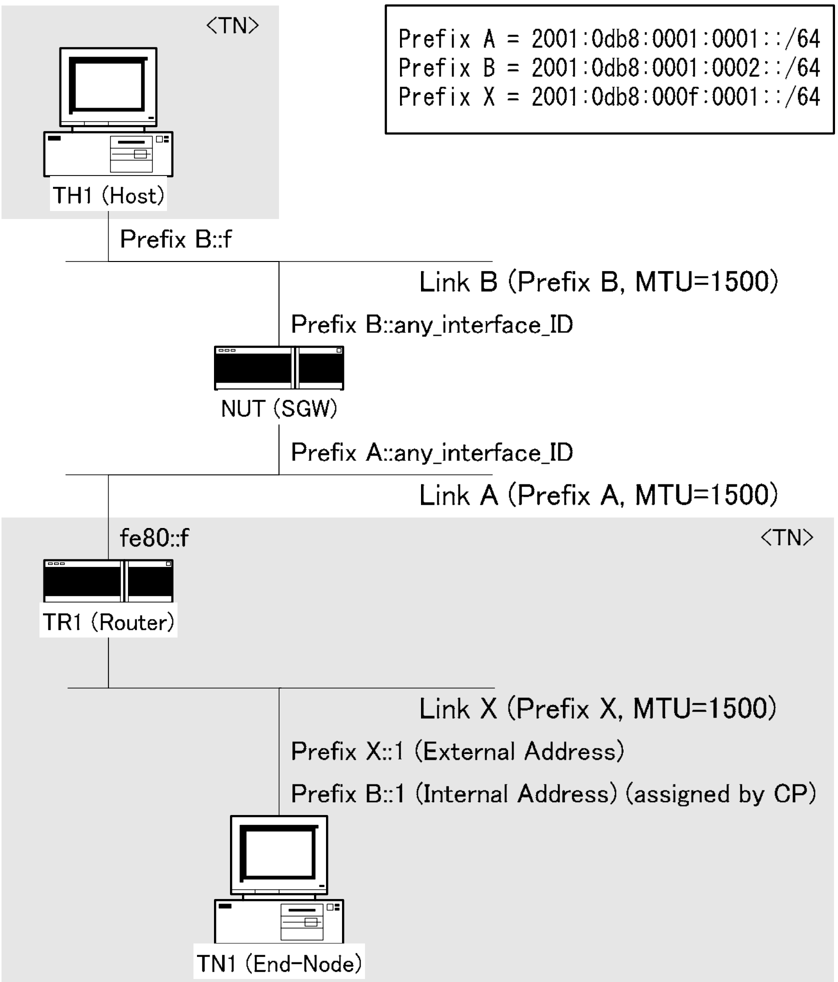
To verify an IKEv2 device properly handles the Initial Exchanges using Pre-shared key

References:

- [RFC 4306] - Sections 2.19 and 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REPLY for INTERNAL_IP6_ADDRESS. Its IPv6 address is Prefix B::1/128. The traffic selector must be configured by the following table. NUT must narrow Traffic Selector to the following address table.

	Traffic Selector

	Source			Destination		
	Address Range	**Next Layer Protocol**	**Port Range**	**Address Range**	**Next Layer Protocol**	**Port Range**
Inbound	TN1 (internal address)	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1 (internal address)	ANY	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                  NUT                  TN1
(Host)               (SGW)             (End-Node)
  |                    |                    |
  |                    |<-------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    |                    | (Packet #1)
  |                    |------------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                    |                    | (Judgment #1)
  |                    |                    |
  |                    |<-------------------| IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                    |                    |                     CP(CFG_REQUEST), SAi2, TSi, TSr})
  |                    |                    | (Packet #2)
  |                    |------------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
  |                    |                    | (Judgment #2)
  |                    |                    |
  |<-------------------+====================| IPsec {Echo Request}
  |                    |                    | (Packet #3) (Judgment #3)
  |--------------------+===================>| IPsec {Echo Reply}
  |                    |                    | (Packet #4) (Judgment #4)
  |                    |                    |
  V                    V                    V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See below

- Packet #2: IKE_AUTH request packet

IPv6 Header	Same as Common Packet #5	
UDP Header	Same as Common Packet #5	
IKEv2 Header	Same as Common Packet #5	
E Payload	Same as Common Packet #5	
IDi Payload	Same as Common Packet #5	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #5	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0
	Payload Length	12
	CFG Type	1 (CFG_REQUEST)
	RESERVED	0
	Configuration Attributes	See below
SA Payload	Same as Common Packet #5	
TSi Payload	Other fields are same as Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #5	

Configuration Attributes	Reserved	0
	Attribute Type	INTERNAL_IP6_ADDRESS
	Length	0

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	::
	Ending Address	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

- Packet #3: Echo Request packet

IPv6 Header	Same as Common Packet #22	
ESP	Same as Common Packet #22	
IPv6 Header	Source Address	Prefyx B::1
	Destination Address	Prefix B::f
ICMPv6 Header	Same as Common Packet #22	

- Packet #4: Echo Reply packet

IPv6 Header	Source Address	Prefyx B::f
	Destination Address	Prefix B::1
ICMPv6 Header	Same as Common Packet #26	

Part A (ADVANCED)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to TH1.
6. Observe the messages transmitted on Link A.
7. TH1 transmits an Echo Reply to TN1.
8. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including “ENCR_3DES”, “AUTH_HMAC_SHA1_96” and “No Extended Sequence Numbers” as proposed algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request to the TH1.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TN1 can send Echo Reply to TH1 instead of sending Echo Request.

Test IKEv2.SGW.R.2.1.2.3: Non zero RESERVED fields in Configuration Payload

Purpose:

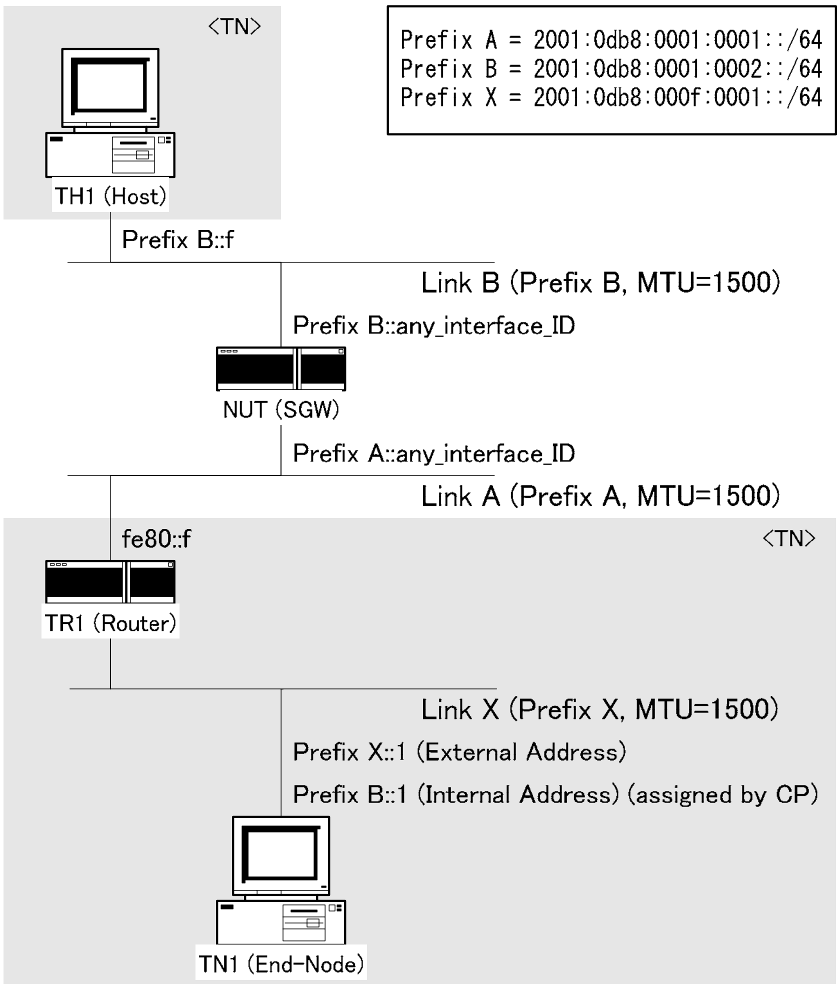
To verify an IKEv2 device ignores the content of RESERVED filed in IKE messages.

References:

- [RFC 4306] - Sections 2.5

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REPLY for INTERNAL_IP6_ADDRESS. Its IPv6 address is Prefix B::1/128. The traffic selector must be configured by the following table. NUT must narrow Traffic Selector to the following address table.

	Traffic Selector					
	Source			**Destination**		
	Address Range	**Next Layer Protocol**	**Port Range**	**Address Range**	**Next Layer Protocol**	**Port Range**
Inbound	TN1 (internal address)	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1 (internal address)	ANY	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1
(Host)             (SGW)            (End-Node)
 |                  |                  |
 |                  |<-----------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
 |                  |                  | (Packet #1)
 |                  |----------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
 |                  |                  | (Judgment #1)
 |                  |                  |
 |                  |<-----------------| IKE_AUTH request (HDR, SK {IDi, AUTH,
 |                  |                  |                     CP(CFG_REQUEST), SAi2, TSi, TSr})
 |                  |                  | (Packet #2)
 |                  |----------------->| IKE_AUTH response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
 |                  |                  | (Judgment #2)
 |                  |                  |
 V                  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below

- Packet #2: IKE_AUTH request packet

IPv6 Header	Same as Common Packet #5	
UDP Header	Same as Common Packet #5	
IKEv2 Header	Same as Common Packet #5	
E Payload	Same as Common Packet #5	
IDi Payload	Same as Common Packet #5	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #5	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	1
	Payload Length	12
	CFG Type	1 (CFG_REQUEST)
	RESERVED	1
	Configuration Attributes	See below
SA Payload	Same as Common Packet #5	
TSi Payload	Other fields are same as Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #5	

Configuration Attributes	Reserved	1
	Attribute Type	INTERNAL_IP6_ADDRESS
	Length	0

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)

	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	::
	Ending Address	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Part A (ADVANCED)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Possible Problems:

- None.

Test IKEv2.SGW.R.2.1.2.4: No Configuration payload

Purpose:

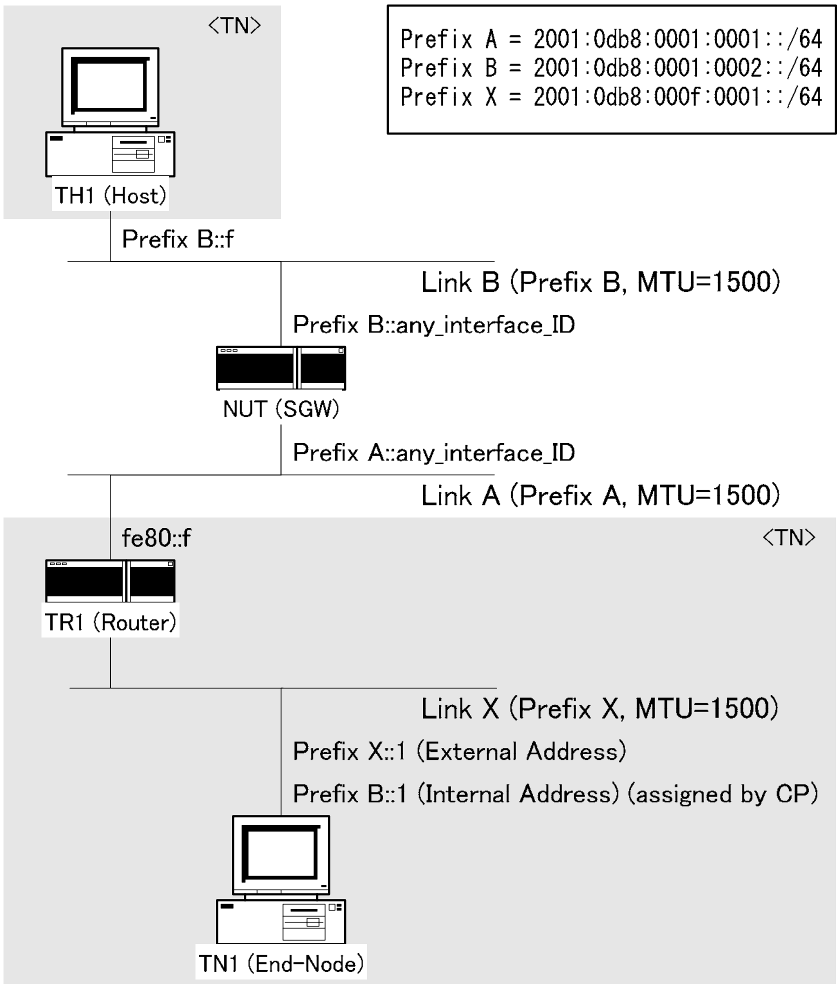
To verify an IKEv2 device properly handles the message which does not include Configuration payload, when the device expects Configuration payload.

References:

- [RFC 4306] - Sections 2.19 and 3.10.1

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REPLY for INTERNAL_IP6_ADDRESS. Its IPv6 address is Prefix B::1/128. The traffic selector must be configured by the following table. NUT must narrow Traffic Selector to the following address table.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1 (internal address)	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1 (internal address)	ANY	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 NUT                  TN1
(SGW)             (End-Node)
  |                    |
  |<-------------------| IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                    | (Packet #1)
  |------------------->| IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                    | (Judgment #1)
  |                    |
  |<-------------------| IKE_AUTH request (HDR, SK {IDi, AUTH, SAi2, TSi, TSr})
  |                    | (Packet #2)
  |------------------->| IKE_AUTH response (HDR, SK {N(FAILED_CP_REQUIRED)})
  |                    | (Judgment #2)
  |                    |
  V                    V
```

Packet #1	See Common Packet #1
Packet #2	See Common Packet #5 This packet does not include CP payload.

Part A (ADVANCED)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including “ENCR_3DES”, “PRF_HMAC_SHA1”, “AUTH_HMAC_SHA1_96” and “D-H group 2” as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response with a Notify payload of type FAILED_CP_REQUIRED.

Possible Problems:

- None.

Test IKEv2.SGW.R.2.1.2.5: Receipt of Multiple CFG_REQUEST

Purpose:

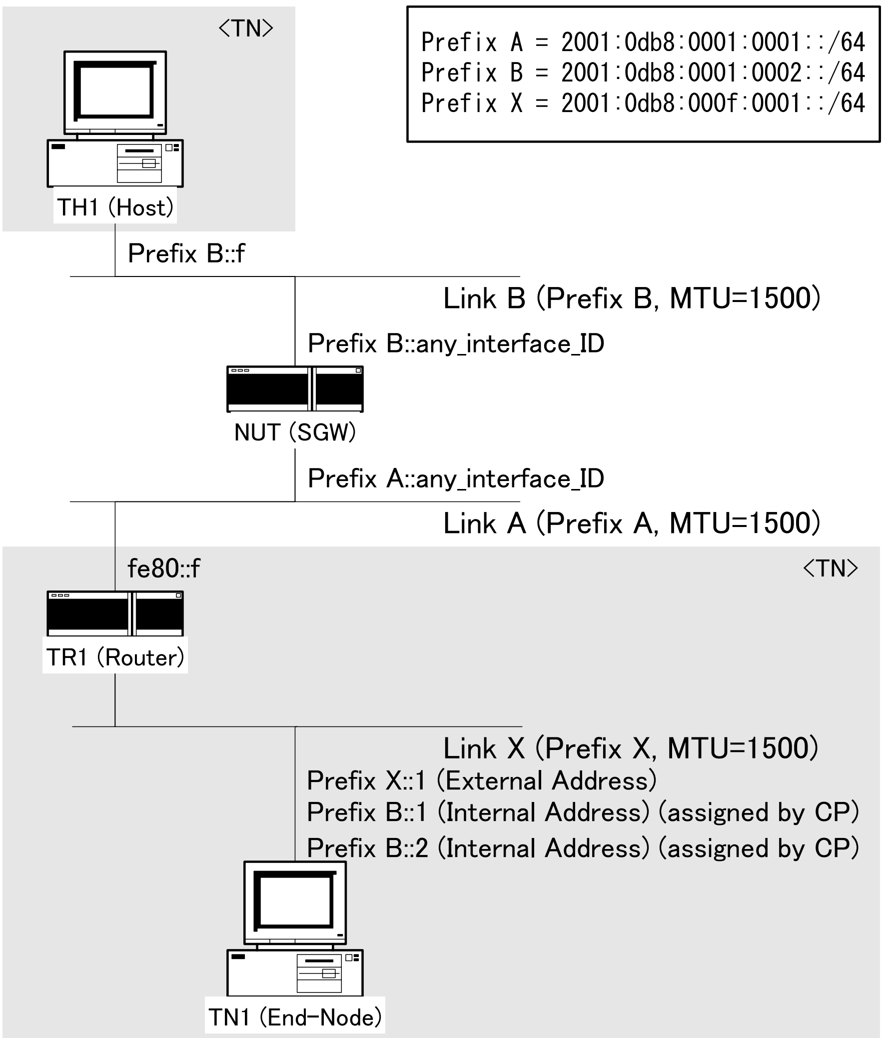
To verify an IKEv2 device properly handles multiple CFG_REQUEST.

References:

- [RFC 4306] - Sections 2.19 and 3.15

Test Setup:

- Network Topology
  Connect the devices according to the following topology.



- Configuration
  In each part, configure NUT according to the Common Configuration except the traffic selector. Configure NUT to transmit CFG_REPLY for INTERNAL_IP6_ADDRESS. Its IPv6 address is Prefix B::1/128. The traffic selector must be configured by the following table. NUT must narrow Traffic Selector to the following address table.

	Traffic Selector					
	Source			Destination		
	Address Range	Next Layer Protocol	Port Range	Address Range	Next Layer Protocol	Port Range
Inbound	TN1 (internal address)	ANY	ANY	Link B	ANY	ANY
Outbound	Link B	ANY	ANY	TN1 (internal address)	ANY	ANY

- Pre-Sequence and Cleanup Sequence
  IKEv2 on the NUT is disabled after each part.

Procedure:

```
 TH1                NUT                TN1
(Host)             (SGW)           (End-Node)
  |                  |                  |
  |                  |<-------------|  IKE_SA_INIT request (HDR, SAi1, KEi, Ni)
  |                  |                  |  (Packet #1)
  |                  |------------->|  IKE_SA_INIT response (HDR, SAr1, KEr, Nr)
  |                  |                  |  (Judgment #1)
  |                  |                  |
  |                  |<-------------|  IKE_AUTH request (HDR, SK {IDi, AUTH,
  |                  |                  |                            CP(CFG_REQUEST), SAi2, TSi, TSr})
  |                  |                  |  (Packet #2)
  |                  |------------->|  IKE_AUTH response (HDR, SK {IDr, AUTH, CP(CFG_REPLY), SAr2, TSi, TSr})
  |                  |                  |  (Judgment #2)
  |                  |                  |
  |<-------------+==============|  IPsec {Echo Request}
  |                  |                  |  (Packet #3) (Judgment #3)
  |--------------+==============>|  IPsec {Echo Reply}
  |                  |                  |  (Packet #4) (Judgment #4)
  |                  |                  |
  |<-------------+==============|  IPsec {Echo Request}
  |                  |                  |  (Packet #5) (Judgment #5)
  |--------------+==============>|  IPsec {Echo Reply}
  |                  |                  |  (Packet #6) (Judgment #6)
  |                  |                  |
  V                  V                  V
```

Packet #1	See Common Packet #1
Packet #2	See below
Packet #3	See below
Packet #4	See below
Packet #5	See below
Packet #6	See below

- Packet #2: IKE_AUTH request packet

IPv6 Header	Same as Common Packet #5	
UDP Header	Same as Common Packet #5	
IKEv2 Header	Same as Common Packet #5	
E Payload	Same as Common Packet #5	
IDi Payload	Same as Common Packet #5	
AUTH Payload	Next Payload	47 (CP)
	Other fields are same as Common Packet #5	
CP Payload	Next Payload	33 (SA)
	Critical	0
	Reserved	0

	Payload Length	16
	CFG Type	1 (CFG_REQUEST)
	RESERVED	0
	Configuration Attributes	See below
SA Payload	Same as Common Packet #5	
TSi Payload	Other fields are same as Common Packet #5	
	Traffic Selectors	See below
TSr Payload	Same as Common Packet #5	

Configuration Attributes	Reserved	0
	Attribute Type	INTERNAL_IP6_ADDRESS
	Length	0
Configuration Attributes	Reserved	0
	Attribute Type	INTERNAL_IP6_ADDRESS
	Length	0

Traffic Selector	TS Type	8 (IPV6_ADDR_RANGE)
	IP Protocol ID	0 (any)
	Selector Length	40
	Start Port	0
	End Port	65535
	Starting Address	::
	Ending Address	ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

- Packet #3: Echo Request packet

IPv6 Header	Same as Common Packet #22	
ESP	Same as Common Packet #22	
IPv6 Header	Source Address	Prefyx B::1
	Destination Address	Prefix B::f
ICMPv6 Header	Same as Common Packet #22	

- Packet #4: Echo Reply packet

IPv6 Header	Source Address	Prefyx B::f
	Destination Address	Prefix B::1
ICMPv6 Header	Same as Common Packet #26	

- Packet #5: Echo Request packet

IPv6 Header	Same as Common Packet #22	
ESP	Same as Common Packet #22	
IPv6 Header	Source Address	Prefyx B::2
	Destination Address	Prefix B::f
ICMPv6 Header	Same as Common Packet #22	

- Packet #6: Echo Reply packet

IPv6 Header	Source Address	Prefyx B::f
	Destination Address	Prefix B::2
ICMPv6 Header	Same as Common Packet #26	

Part A (ADVANCED)

1. TN1 transmits an IKE_SA_INIT request to NUT.
2. Observe the messages transmitted on Link A.
3. TN1 transmits an IKE_SA_INIT request to the NUT.
4. Observe the messages transmitted on Link A.
5. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to TH1.

6. Observe the messages transmitted on Link A.
7. TH1 transmits an Echo Reply to TN1.
8. Observe the messages transmitted on Link B.
9. TN1 transmits an Echo Request with IPsec ESP using corresponding algorithms to TH1.
10. Observe the messages transmitted on Link A.
11. TH1 transmits an Echo Reply to TN1.
12. Observe the messages transmitted on Link B.

Observable Results:

Part A

Step 2: Judgment #1
The NUT transmits an IKE_SA_INIT response including "ENCR_3DES", "PRF_HMAC_SHA1", "AUTH_HMAC_SHA1_96" and "D-H group 2" as proposed algorithms.

Step 4: Judgment #2
The NUT transmits an IKE_AUTH response including "ENCR_3DES", "AUTH_HMAC_SHA1_96" and "No Extended Sequence Numbers" as proposed algorithms.

Step 6: Judgment #3
The NUT forwards an Echo Request to the TH1.

Step 8: Judgment #4
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Step 10: Judgment #5
The NUT forwards an Echo Request to the TH1.

Step 12: Judgment #6
The NUT forwards an Echo Reply with IPsec ESP using corresponding algorithms.

Possible Problems:

- Because the destination address of Echo Request is the TN itself, TN may respond to Echo Request automatically. In that case, TN1 can send Echo Reply to TH1 instead of sending Echo Request.