IP Security

Ferucio Laurenţiu Ţiplea

Department of Computer Science
“AL.I.Cuza” University of Iaşi
Iaşi, Romania

December 14, 2016

Outline

1. What is IPsec?

2. Transport and Tunnel Modes

3. More on AH and ESP
   - AH format
   - ESP format

4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases

5. Internet key exchange

Outline

1. What is IPsec?
2. Transport and Tunnel Modes
3. More on AH and ESP
   - AH format
   - ESP format
4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases
5. Internet key exchange

Security Issues with IP

S. Bellovin: *Security Problems in the TCP/IP Protocol Suite*, Computer Communication Review, Vol. 19, No. 2, pp. 32-48, April 1989

- Eavesdropping (sniffing, snooping)
- Data modification
- Sequence number spoofing
- IP address spoofing
- Routing attacks

The Internet will never be fully secure ...

IPsec: What Is It ?

- Security architecture for the Internet Protocol (IPv4 and IPv6)
- Provides security services at the IP layer
- Provides security in three situations
  - host – host
  - host – security gateway
  - security gateway – security gateway
- Operates in two modes
  - transport (for end-to-end)
  - tunnel (for VPN)

IPsec: Networking Concepts

- Node
  - device attached to a network where messages can be created, received, or transmitted
  - examples: computers, personal digital assistants (PDAs), cell phones, or various other networked devices
  - on a TCP/IP network, a node is any device with an IP address
- Host : node that is a computer
- Security gateway
  - system that implements IPsec protocols
  - examples: router or firewall implementing IPsec

IPsec: Fundamental Components

1. Security protocols
   - Authentication Header (AH) : piece of information associated to an IP datagram in order to authenticate certain fields of the datagram
   - Encapsulating Security Payload (ESP) : obtained from an IP datagram by encrypting, and optionally authenticating, certain fields of the datagram
2. Security associations
3. Key management protocols
4. Algorithms for authentication and encryption

Because of these protocols are provided at the IP layer, they can be used by any higher layer protocol (e.g., TCP, UDP, ICMP etc.)

IPsec Security Services

Security service	AH	ESP	ESP with auth
access control	yes	yes	yes
data integrity	yes		yes
data origin authentication	yes		yes
confidentiality		yes	yes
rejection of replayed packages	yes		yes
limited traffic flow confidentiality		yes	yes

Outline

1. What is IPsec?

2. Transport and Tunnel Modes

3. More on AH and ESP
   - AH format
   - ESP format

4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases

5. Internet key exchange

IP Datagrams

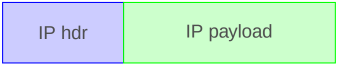


Figure: IPv4 datagram

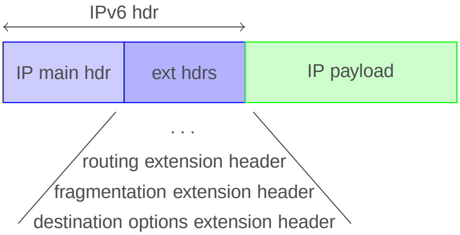

IPv6 hdr
IP main hdr
ext hdrs
IP payload
. . .
routing extension header
fragmentation extension header
destination options extension header

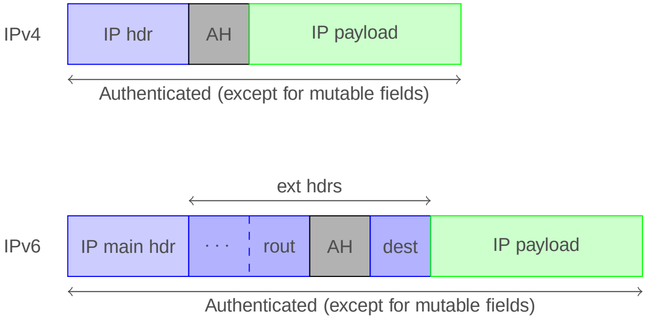
Figure: IPv6 datagram

Transport Mode

- Typically, the transport mode is used for communication between two hosts (e.g., a client and a server or two workstations)
- Gateways are not required to support the transport mode. A gateway is allowed to support the transport mode when it acts as a host, that is, when the traffic is destined to the gateway itself
- Due to its definitions, the transport mode provides protection for upper layer protocols (e.g., TCP or UDP)
- ☺ Fewer processing costs
- ☹ Mutable fields are not authenticated

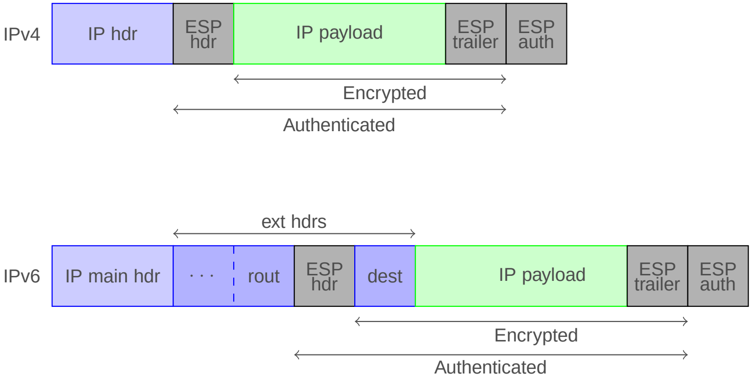
AH in Transport Mode

In the transport mode, AH authenticates the IP payload and selected portions of the IP header (e.g., mutable and unpredictable fields are not authenticated)


IPv4
IP hdr
AH
IP payload
Authenticated (except for mutable fields)
ext hdrs
IPv6
IP main hdr
· · ·
rout
AH
dest
IP payload
Authenticated (except for mutable fields)

ESP in Transport Mode

In the transport mode, ESP encrypts and optionally authenticates the IP payload (but not the IP header)

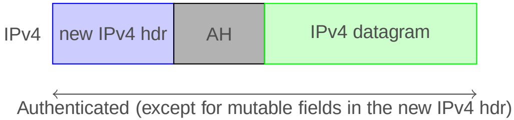


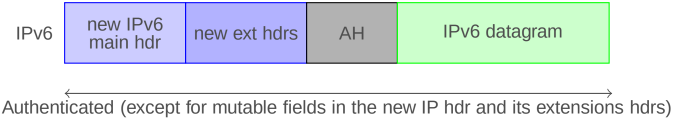
Tunnel Mode

- Tunneling means encapsulation and it consists of wrapping a packet in a new one
- Tunnel mode is used whenever either end of an SA is a security gateway:
  - host – security gateway
  - security gateway – security gateway (such as two firewalls)
  - security gateway – host
- Remark that hosts must support both transport and tunnel mode
- ☺ Total protection (possibility of using private addresses)
- ☹ Extra processing costs

AH in Tunnel Mode

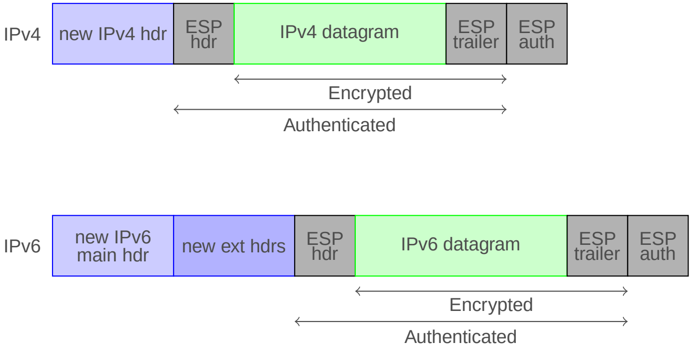
In the tunnel mode, AH authenticates the entire inner IP packet plus selected portions of the outer IP header and outer IP extension headers


IPv4
new IPv4 hdr
AH
IPv4 datagram
Authenticated (except for mutable fields in the new IPv4 hdr)


IPv6
new IPv6 main hdr
new ext hdrs
AH
IPv6 datagram
Authenticated (except for mutable fields in the new IP hdr and its extensions hdrs)

ESP in Tunnel Mode

In the tunnel mode, ESP (with authentication) encrypts (and authenticates) the inner IP packet



Outline

1 What is IPsec?

2 Transport and Tunnel Modes

3 More on AH and ESP
- AH format
- ESP format

4 Security Associations
- Security associations
- Basic combinations of SAs
- Security association and policy databases

5 Internet key exchange

Outline

1. What is IPsec?

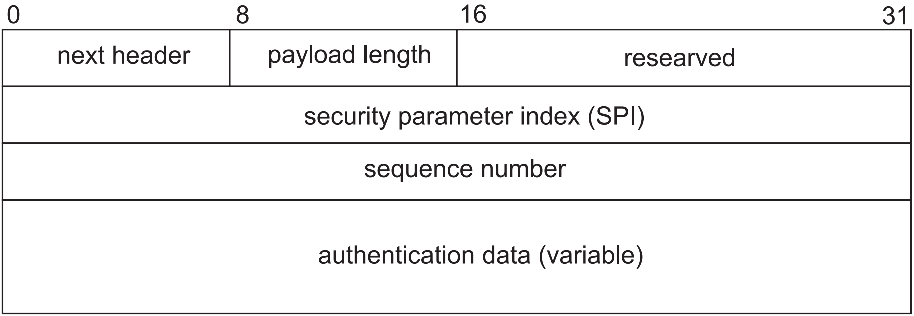
2. Transport and Tunnel Modes

3. More on AH and ESP
   - AH format
   - ESP format

4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases

5. Internet key exchange

Authentication Header



0 8 16 31

next header

payload length

researved

security parameter index (SPI)

sequence number

authentication data (variable)

Figure: AH format

Authentication Header

- Sequence number field : designed to thwart reply attacks
- Authentication data field : contains the Integrity Check Value (ICV), or MAC, for the packet.

  RFC 4835 recommendation:

Requirement	Authentication algorithm
MUST	HMAC-SHA-1-96
SHOULD+	AES-XCBC-MAC-96
MAY	HMAC-MD5-95

- Source Address and Destination Address are always authenticated under AH and ESP and, therefore, address spoofing is prevented

Outline

1 What is IPsec?

2 Transport and Tunnel Modes

3 More on AH and ESP
- AH format
- ESP format

4 Security Associations
- Security associations
- Basic combinations of SAs
- Security association and policy databases

5 Internet key exchange

Encapsulating Security Payload Format

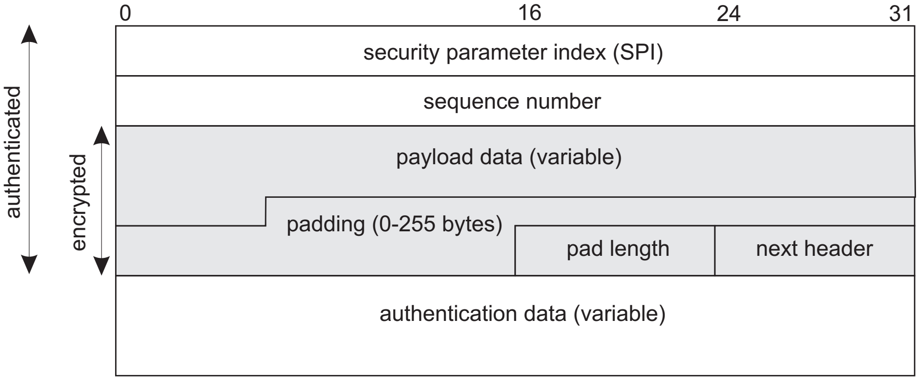


Figure: ESP format

Encryption in ESP

RFC 4835 recommendation:

Requirement	Encryption algorithm
MUST	NULL
MUST	AES-CBC with 128-bit keys
MUST-	3DES-CBC
SHOULD	AES-CTR
SHOULD NOT	DES-CBC

NULL does nothing to alter data

Authentication in ESP

RFC 4835 recommnedation:

Requirement	Authentication algorithm
MUST	HMAC-SHA-1-96
SHOULD+	AES-XCBC-MAC-96
MAY	NULL
MAY	HMAC-MD5-95

Authentication and encryption can each be "NULL", but not at the same time

Outline

1. What is IPsec?
2. Transport and Tunnel Modes
3. More on AH and ESP
   - AH format
   - ESP format
4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases
5. Internet key exchange

Outline

1. What is IPsec?

2. Transport and Tunnel Modes

3. More on AH and ESP
   - AH format
   - ESP format

4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases

5. Internet key exchange

Security Associations

A security association (SA) is a unidirectional logical connection between two IP systems, uniquely identified by a triple

$$(SPI,\ IP\ destination\ address,\ security\ protocol)$$

where

- SPI (security parameter index) is a 32-bit value used to identify different SAs with the same destination address and the same security protocol
- IP destination address can be unicast, broadcast, or multicast
- security protocol – this can be either AH or ESP

Security Associations

1. SAs are uniderectional ! Thus, for bidirectional communication bewteen two IPsec systems there must be two SAs definied, one for each direction
2. A single SA gives security to the traffic carried by it either by using AH or ESP, but not both
3. For a connection that needs to be protected by both AH and ESP, two SAs must be defined for each direction

Outline

1. What is IPsec?
2. Transport and Tunnel Modes
3. More on AH and ESP
   - AH format
   - ESP format
4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases
5. Internet key exchange

SA Bundle

- An SA bundle is a sequence of SAs through which traffic must be processed to provide a desired security
- SAs may be combined into bundles in two ways:
  - transport adjacency – consists of applying in the transport mode both security protocols to the same IP datagram
  - iterated tunneling – consists of applying multiple layers of security protocols through IP tunneling (although there is no limit in the nesting levels, more than three levels is considered impractical)

These approaches can be combined: e.g., an IP packet with transport adjacency IPsec headers can be sent through nested tunnels

End-to-end Security

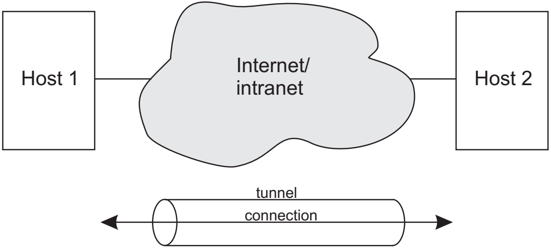


Figure: End-to-end security

Two hosts are connected through the Internet or an intranet without any security gateway between them. They can use ESP, AH, or both. Either transport or tunnel mode can be applied

Basic VPN Support

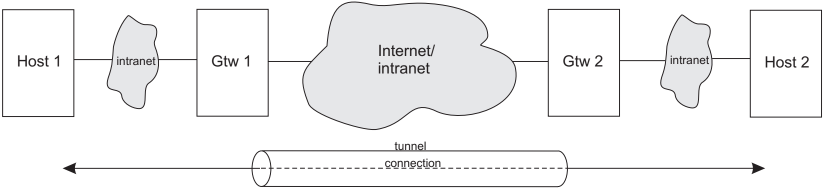


Figure: Basic VPN support

The hosts in the intranets are not required to support IPsec, but the gateways are required to run IPsec and support tunnel mode (either with AH or ESP)

End-to-end Security with VPN Support

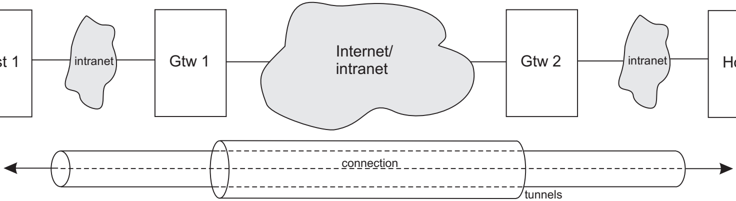


Figure: End-to-end security with VPN support

This is a combination of the previous two cases. For instance, the gateways may use AH in tunnel mode, while the hosts use ESP in transport mode

Remote Access

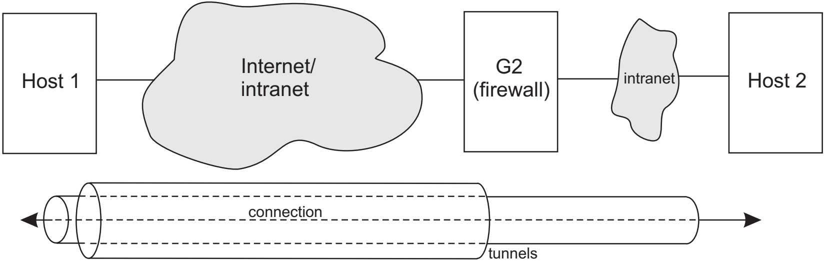


Figure: Remote access

Between the host H1 and the firewall G2, only the tunnel mode is required (e.g., AH in tunnel mode), and between the host H1 and H2, either transport or tunnel mode can be used (e.g., ESP in transport mode)

Outline

1. What is IPsec?
2. Transport and Tunnel Modes
3. More on AH and ESP
   - AH format
   - ESP format
4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases
5. Internet key exchange

SAD and SPD

1. Each SA has an entry in a Security Association Database (SAD)

2. A Security Policy Database (SPD) specifies what services are to be offered to IP datagrams and in what fashion

3. An SPD consists of an ordered lists of policy entries, each policy being keyed by one or more (traffic) selectors that define the set of IP traffic encompassed by this policy entry

4. Example of policy entry: all matching traffic must be protected by ESP in transport mode using 3DES-CBC with an explicit IV, nested inside of AH in tunnel mode using HMAC-SHA-1

5. SPD must be consulted during the processing of all traffic (inbound or outbound), including non-IPsec traffic

Outline

1. What is IPsec?
2. Transport and Tunnel Modes
3. More on AH and ESP
   - AH format
   - ESP format
4. Security Associations
   - Security associations
   - Basic combinations of SAs
   - Security association and policy databases
5. Internet key exchange

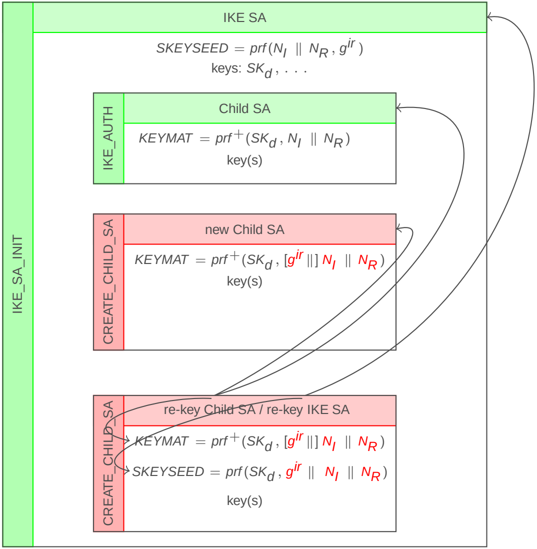
Internet Key Exchange

- Internet Key Exchange (IKE) is a component of IPsec that
  - establishes an IKE SA that includes shared secrets
  - performs mutual authentication between parties
  - establishes AH and ESP SAs and a set of cryptographic algorithms to be used by them
- The design of IKE was influenced by three protocols:
  - STS (Station-to-Station) protocol – this was discussed in one of our previous lectures
  - SKEME protocol – developed specifically for IPsec, SKEME is an extension of Photuris suggested in 1996 by H. Krawczyk
  - Oakley protocol – this is a key-agreement protocol proposed by H. Orman in 1998 (RFC 2412). It allows authenticated parties to exchange keying material across an insecure connection

IKE Exchanges

- Exchange : pair of messages consisting of a request and a response
- Types of exchanges in IKE:
  - The first exchange (IKE_SA_INIT)
    - negotiates security parameters for the IKE SA
    - sends nonces
    - sends DH values
  - The second exchange (IKE_AUTH)
    - transmits identities
    - proves knowledge of the secrets corresponding to the two identities
    - sets up an SA for the first (and often only) AH or ESP Child SA
  - Subsequent exchanges:
    - CREATE_CHILD_SA : creates new Child SAs or re-keys (create a new SA and then delete the old SA) both IKE SAs and Child SAs
    - INFORMATIONAL : deletes an SA, reports error conditions, or does other housekeeping

IKE Exchanges


IKE_SA_INIT
IKE SA
$SKEYSEED = prf(N_I \parallel N_R, g^{ir})$
keys: $SK_d, \ldots$
IKE_AUTH
Child SA
$KEYMAT = prf^{+}(SK_d, N_I \parallel N_R)$
key(s)
CREATE_CHILD_SA
new Child SA
$KEYMAT = prf^{+}(SK_d, [g^{ir} \parallel] N_I \parallel N_R)$
key(s)
CREATE_CHILD_SA
re-key Child SA / re-key IKE SA
$KEYMAT = prf^{+}(SK_d, [g^{ir} \parallel] N_I \parallel N_R)$
$SKEYSEED = prf(SK_d, g^{ir} \parallel N_I \parallel N_R)$
key(s)

IKE_SA_INIT

IKE_SA_INIT
$I \rightarrow R: \quad Hdr, SA_{I_1}, KE_I, N_I$
$R \rightarrow I: \quad Hdr, SA_{R_1}, KE_R, N_R\ [, CertReq]$

where:

- Hdr contains SPIs, version numbers, and flags
- SA_{I_1} states the cryptographic algorithms the initiator supports for the IKE SA
- SA_{R_1} is the responder choice selected from the initiator's offered choices (SA_{I_1})
- N_I and N_R are nonces
- KE_I and KE_R are DH values (g^i and g^r)
- $CertReq$: certificate request

IKE_SA_INIT: Key Generation

At this point, each party can generate all keys for IKE SA:

$$\begin{array}{lcl} SKEYSEED & = & prf(N_I \parallel N_R, g^{ir}) \\ KEYS & = & prf^{+}(SKEYSEED, N_I \parallel N_R \parallel SPI_I \parallel SPI_R) \\ KEYS & = & SK_d \parallel SK_{ai} \parallel SK_{ar} \parallel SK_{ei} \parallel SK_{er} \parallel SK_{pi} \parallel SK_{pr} \parallel \cdots \end{array}$$

where:

- prf is a psudo-random function
- prf^{+} is an iteration of prf

$$\begin{array}{lcl} prf^{+}(K, T_0) & = & T_1 \parallel T_2 \parallel T_3 \parallel \cdots \\ T_1 & = & prf(K, T_0 \parallel 0x01) \\ T_2 & = & prf(K, T_1 \parallel 0x02) \\ \ldots & & \end{array}$$

In what follows, $\{\cdot\}_{SK}$ means authenticated encryption by $SK_{ex} + SK_{ax}$, with $x \in \{i, r\}$

IKE_AUTH

IKE_AUTH

$I \rightarrow R: \quad Hdr, \{ID_I, [Cert,][CertReq,][ID_R,]Auth, SA_{I_2}, TS_I, TS_R\}_{SK}$

$R \rightarrow I: \quad Hdr, \{ID_R, [Cert,]Auth, SA_{R_2}, TS_I, TS_R\}_{SK}$

where:

- ID_I, ID_R : identities
- $Auth$: authentication payload (based on SK_{pi} and SK_{pr})
- $Cert$: certificate payload
- SA_{I_2}, SA_{R_2} : the initiator begins negotiation of a Child SA using the SA_{I_2} payload, and the receptor completes the negotiation with SA_{R_2}
- TS_I, TS_R : traffic selectors
    - A traffic selector is a list of IP addresses and port numbers that are to be protected by the SA
    - TS_I (TS_R) specifies source (destination) addresses and ports

IKE_AUTH: Key Generation for Child SA

When the first Child SA is created by IKE_AUTH, the keys are generated as follows:

- The keying material is

$$KEYMAT = prf^{+}(SK_d, N_I \parallel N_R)$$

  where N_I and N_R are the nonces from the IKE_SA_INIT exchange

- Generally, keys are taken from *KEYMAT* in the order: encryption key and then integrity key

CREATE_CHILD_SA

Used to:

- Create new Child SA (recall that the first Child SA is created by IKE_AUTH)
- Re-key a Child SA
- Re-key an IKE SA – the main reason for rekeying the IKE SA is to ensure that the compromise of old keying material does not provide information about the current keys, or vice versa

Re-keying an SA: create a new SA and then delete the old one

CREATE_CHILD_SA: New Child SA

CREATE_CHILD_SA: New Child SA	
$I \rightarrow R$:	$Hdr, \{SA, N_I[, KE_I], TS_I, TS_R\}_{SK}$
$R \rightarrow I$:	$Hdr, \{SA, N_R[, KE_R], TS_I, TS_R\}_{SK}$

where:

- SA : the new security association the initiator wants to create
- If KE_I and KE_R are not used, the keys are generated as in the case of a Child SA created by IKE_SA but with the fresh nonces N_I and N_R
- If KE_I and KE_R are used, the keys are generated as follows:
  - $KEYMAT \; = \; prf^+(SK_d, g^{ir} \parallel N_I \parallel N_R)$ (g^{ir}, N_I, N_R are the fresh ones)
  - the same rules for taking the keys

CREATE_CHILD_SA: Re-keying a Child SA

CREATE_CHILD_SA: Re-keying a Child SA
$I \rightarrow R: \quad Hdr, \{N(REKEY\_SA), SA, N_I[, KE_I], TS_I, TS_R\}_{SK}$
$R \rightarrow I: \quad Hdr, \{SA, N_R[, KE_R], TS_I, TS_R\}_{SK}$

where:

- N(REKEY_SA) identifies (by the SPI field) the SA to be rekeyed
- The keys are generated as in the case of creation of a new Child SA

CREATE_CHILD_SA: Re-keying IKE SA

CREATE_CHILD_SA: Re-keying IKE SA
$I \rightarrow R: \quad Hdr, \{SA, N_I, KE_I\}_{SK}$
$R \rightarrow I: \quad Hdr, \{SA, N_R, KE_R\}_{SK}$

where:

- SA re-keys the current IKE SA
- The new SKEYSEED is computed by

  $$SKEYSEED = prf(SK_d, g^{ir} \parallel N_I \parallel N_R)$$

  where SK_d and prf are the old ones
- The new SK_d, SK_{ai} etc., are computed as usual (a new prf may be used)

INFORMATIONAL

INFORMATIONAL
$I \rightarrow R: \quad Hdr, \{[N,]\,[D,]\,[CP,]\ldots\}_{SK}$
$R \rightarrow I: \quad Hdr, \{[N,]\,[D,]\,[CP,]\ldots\}_{SK}$

where:

- N : notify
- D : delete
- CP : configuration