Functional Package for IPsec

Version: 1.0
2022-01-06
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2022-01-06	Start of first draft.

Contents

1 Introduction
1.1 Overview
1.2 Terms
1.2.1 Common Criteria Terms
1.2.2 Technical Terms
1.3 Compliant Targets of Evaluation
1.4 Use Cases
2 Conformance Claims
3 Security Functional Requirements
3.1 Auditable Events for Mandatory SFRs
3.2 Cryptographic Support (FCS)
3.3 Identification and Authentication (FIA)
Appendix A - Implementation-Dependent Requirements
Appendix B - Use Case Templates
B.1 IPsec Endpoint
B.2 EAP
B.3 Pre-Shared Keys
Appendix C - Acronyms
Appendix D - Bibliography

1 Introduction

1.1 Overview

Internet Protocol Security (IPsec) is a suite of open standards for ensuring private communications over public networks. It is typically used to encrypt Internet Protocol (IP) traffic between hosts in a network and to create a virtual private network (VPN). A VPN is a virtual network built on top of existing physical networks that provides a secure communications mechanism for data and control information transmitted between computers or networks. IPsec can also be used as a component that provides security for other internet protocols, such as the User Datagram Protocol (UDP).

The main components of IPsec are Encapsulating Security Protocol (ESP) and Internet Key Exchange (IKE). ESP is the protocol used to to transport encrypted and integrity-protected communications across the network. IKE is the protocol used to set up and manage IPsec connections.

This *Functional Package for IPsec* provides a collection of requirements and evaluation activities for IPSec implementations. The intent of this package is to provide PP, cPP, and PP-Module authors with a readily consumable collection of SFRs and EAs to be integrated into their documents. This Package can be used to to evaluate the IPsec functionality of TOEs that are not themselves VPN clients. For example, it could be used to evaluate the trusted channel functionality of an operating system that chooses to use IPsec rather than SSH or TLS to implement secure remote management. And of course, this Package could be used to encapsulate the IPsec-specific requirements in a VPN technology evaluation as well.

As such, this Package attempts to specify only requirements and evaluation activities for IPsec implementations as distinct from those for VPN implementations, such as VPN gateways and clients.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Extended Package (EP)	A deprecated document form for collecting SFRs that implement a particular protocol, technology, or functionality. See Functional Packages.
Functional Package (FP)	A document that collects SFRs for a particular protocol, technology, or functionality.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.

Configuration (PP-Configuration)	
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.

1.2.2 Technical Terms

Certificate Authority (CA)	An entity that issues digital certificates.
Certificate Signing Request (CSR)	A message sent from an applicant to a registration authority of the public key infrastructure in order to apply for a digital identity certificate.
Distinguished Name (DN)	A field in an X.509 certificate that uniquely identifies a person, organization, or business.
Eliptic Curve group modulo a Prime (ECP)	Eliptic Curve Group Modulo a Prime.
Encapsulating Security Payload (ESP)	The protocol used by IPsec to transport encrypted and integrity-protected communications across the network.
Extended Authentication (XAUTH)	An authentication scheme that supports an additional level of authentication by allowing the IPSec gateway to request extended authentication from remote users.
Extended Sequence Number (ESN)	An extension to the standard that allows IPsec to use 64-bit sequence numbers.
Extensible Authentication Protocol (EAP)	A framework for adding arbitrary authentication methods in a standardized way to any protocol. The most common EAP method used with IKEv2 is EAP-TLS.
Fully Qualified Domain Name (FQDN)	A domain name that specifies its exact location in the hierarchy of the Domain Name System (DNS).
Internet Control Message Protocol (ICMP)	A supporting protocol in the Internet Protocol suite. It is used by network devices to send error messages and operational information indicating success or failure when communicating with another IP address.
Internet Key Exchange (IKE)	The protocol used by IPsec to set up and manage IPsec connections. This includes negotiating IPsec connection settings, authenticating endpoints to each other, defining the security parameters of IPsec-protected connections, and negotiating session keys. IKEv2 is the current version.

Internet Protocol Security (IPsec)	A suite of open standards for ensuring private communications over public networks.
Internet Security Association and Key Management Protocol (ISAKMP)	A protocol defined by RFC 2408 for establishing Security association (SA) and cryptographic keys in an Internet environment.
Online Certificate Status Protocol (OCSP)	An Internet protocol used for obtaining the revocation status of an X.509 digital certificate.
Pre-Shared Key (PSK)	A secret that was previously shared between two parties before it needs to be used.
Security Association (SA)	The establishment of shared security attributes between two network entities to support secure communication.
Security Policy Database (SPD)	A set of rules that determines whether a packet is subject to IPsec processing. Each entry in the SPD represents a policy that defines how the set of traffic covered under the policy will be processed.
User Datagram Protocol (UDP)	A communications protocol that is primarily used to establish low-latency and loss-tolerating connections between applications on the internet.
Virtual Private Network (VPN)	An extension of a private network across a public or shared network that allows users to exchange data as though they were connected directly to the private network.

1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this Functional Package (FP) is an IT product that includes an implementation of IPsec. Typically this is a VPN client or VPN gateway - for which IPsec functionality is fundamental. But the TOE could also be an operating system or other IT product for which IPsec functionality is ancillary. This FP describes the security functionality of IPsec in terms of [CC].

The contents of this FP must be appropriately incorporated into a PP, cPP, or PP-Module. When this Package is so incorporated, the ST must include selection-based requirements in accordance with the selections or assignments indicated in the incorporating document.

The PP, cPP, or PP-Module that instantiates this Package must typically include the following components in order to satisfy dependencies of this Package. It is the responsibility of the PP, cPP, or PP-Module author who incorporates this FP to ensure that dependence on these components is satisfied, either by the TOE or by assumptions about its Operational Environment.

An ST must identify the applicable version of the PP, cPP, or PP-Module and this Functional Package in its conformance claims.

Component	Explanation
FCS_CKM.1	To support key generation for IPsec, the incorporating document must include FCS_CKM.1 and specify the corresponding algorithms.
FCS_CKM.2	To support key establishment for IPsec, the incorporating document must include FCS_CKM.2 and specify the corresponding algorithms.
FCS_CKM_EXT.5	To support key derivation for IPsec, the incorporating document may need to include FCS_CKM_EXT.5 and specify the corresponding key derivation algorithms.
FCS_COP.1	To support the cryptography needed for IPsec communications, the incorporating document must include FCS_COP.1 (iterating as needed) to specify AES with corresponding key sizes and modes, digital signature generation and verification function (at least one of RSA or ECDSA), a cryptographic hash function, and a keyed-hash message authentication function. In particular, this Package requires that the TOE support AES-GCM-128 and AES-GCM-256 for ESP, and AES-CBC-128 and AES-CBC-256 for IKE.
FCS_RBG_EXT.1	To support random bit generation needed for IPsec key generation, the incorporating document must include a requirement that specifies the TOE's ability to invoke or provide random bit generation services, commonly identified as FCS_RBG_EXT.1.
FIA_X509_EXT.1	To support establishment of IPSec communications using a public key algorithm that includes X.509, the incorporating document must include FIA_X509_EXT.1. This documents requires that X.509v3 certificates be used in IKE peer authentication. If this is pretty much the same across all technologies, then maybe it should be brought in to this package.

FIA_X509_EXT.2	To support establishment of IPsec communications using a public key algorithm that includes X.509, the incorporating document must include FIA_X509_EXT.2 to specify the reasons for using X.509 certificates. But is this really a dependency for this package? I don't think so.
FPT_STM.1	To support establishment of IPSec communications using a public key algorithm that includes X.509, the incorporating document must include FPT_STM.1 or some other requirement that ensures reliable system time. Note however that support for time-based rekey thresholds is selectable and not mandatory.

1.4 Use Cases

This Package supports several use cases. The cases enumerated below change which SFRs in the below collection are mandatory.

[USE CASE 1] IPsec Endpoint

This is the default use case for this package. The mandatory requirement is FCS_IPSEC_EXT.1 and all other SFRs are claimed depending on selections made within this requirement.

For changes to included SFRs, selections, and assignments required for this use case, see B.1 IPsec Endpoint.

[USE CASE 2] EAP

This use case adds physical protections to the base requirements for server-class hardware. Additional physical protections are required because the platform us assumed to be minimally protected by the by the operational environment. This use case can also be invoked for servers in data centers where there are enhanced security requirements.

This use case adds requirements for audit, physical protections, and Administrator authentication to the base mandatory SFRs. It removes the assumption that the TOE is physically protected by the OE.

For changes to included SFRs, selections, and assignments required for this use case, see B.2 EAP.

[USE CASE 3] Pre-Shared Keys

This use case defines the base requirements for portable clients.

This use case adds no requirements to the base mandatory SFRs.

For changes to included SFRs, selections, and assignments required for this use case, see B.3 Pre-Shared Keys.

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to this Package, as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

CC Conformance Claims

This Package is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Revision 5.

PP Claim

This Package does not claim conformance to any Protection Profile.

Package Claim

This Package does not claim conformance to any packages.

3 Security Functional Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

3.1 Auditable Events for Mandatory SFRs

The auditable events specified in this Package are included in a Security Target if the incorporating PP, cPP, or PP-Module supports audit event reporting through FAU_GEN.1 and all other criteria in the incorporating document are met.

Table 1: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_IPSEC_EXT.1	Decisions to DISCARD or BYPASS network packets processed by the TOE.	Presumed identity of source subject. The entry in the SPD that applied to the decision.
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Identity of destination subject. Reason for failure.
FCS_IPSEC_EXT.1	Establishment/Termination of an IPsec SA.	Identity of destination subject. Transport layer protocol, if applicable. Source subject service identifier, if applicable. Non-TOE endpoint of connection (IP address) for both successes and failures.

3.2 Cryptographic Support (FCS)

FCS_EAP_EXT.1 EAP-TLS

This is a selection-based component. Its inclusion depends upon selection from FCS_IPSEC_EXT.1.11, FCS_IPSEC_EXT.1.11.

FCS_EAP_EXT.1.1

The TSF shall implement [**selection**: *EAP-TLS protocol as specified in [RFC 5216], EAP-TTLS as specified in [RFC 5281]*] as updated by [RFC 8996] with TLS implemented using mutual authentication in accordance with the [Functional Package for TLS].

FCS_EAP_EXT.1.2

The TSF shall generate random values used in the [**selection**: *EAP-TLS*, *EAP-TTLS*] exchange using the RBG specified in FCS_RBG_EXT.1.

FCS_EAP_EXT.1.3

The TSF shall support peer authentication using certificates and [**selection**: *PSK*, *HOTP*, *TOTP*, *[**assignment**: other Authentication-verification protocols]*, *no other authentication*] as updated by [RFC 8996] with TLS implemented using mutual authentication in accordance with the [Functional Package for TLS].

FCS_EAP_EXT.1.4

The TSF shall not forward a EAP-success response if the client certificate is not valid according to FIA_X509_EXT.1.

FCS_EAP_EXT.1.5

The TSF shall use the MSK from the [**selection**: *EAP-TLS*, *EAP-TTLS*] response as the IKEv2 shared secret in the authentication payload.

Evaluation Activities

FCS_EAP_EXT.1
TSS
The evaluator shall verify that the TSS describes the use of EAP options for each of the selected peer authentication mechanisms, that TLS with mutual authentication is used, that the random values are from an appropriate source, and that the EAP MSK is derived from the TLS master key and is used as the IKEv2 shared key.
Guidance
The evaluator shall verify that the guidance documents describe any configurable features of the EAP or TLS functionality, including instructions for configuration of the authenticators and registration processes for clients.
Tests
Testing for TLS functionality is in accordance with the TLS package.

For each supported EAP method claimed in FCS_EAP_EXT.1.1 and for each authentication method claimed in FCS_EAP_EXT.1.3, the evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall follow AGD guidance to configure the TSF to use the EAP method claimed. The evaluator shall follow AGD guidance to configure the TSF to use the authentication method claimed and, for EAP-TTLS, register an endpoint with appropriate key material required for the authentication method. The evaluator shall establish an IPsec connection using a test endpoint with a valid certificate and, for EAP-TTLS, configured to provide a correct value for the configured authenticator. The evaluator shall observe that the IPsec connection is successful.*
- ***Test 2:*** *[conditional] If EAP-TTLS is supported, the evaluator shall cause the test endpoint with a valid certificate to send an invalid authenticator for the claimed authentication method:*
  - *For HOTP, replay the HOTP value sent previously.*
  - *For TOTP or PSK, modify a byte of the properly constructed value,and observe that the TSF aborts the connection.*
- ***Test 3:*** *The evaluator shall establish a new, valid certificate for a test endpoint using an identifier not corresponding to a registered user. For EAP-TTLS, the evaluator shall cause the test endpoint using this certificate to send a correct authenticator value for the registered user. The evaluator shall initiate an IPsec connection from the test endpoint to the TSF and observe that the TSF aborts the connection.*
- ***Test 4:*** *The evaluator shall follow AGD guidance to configure the TSF to use a supported EAP method and register the user with key material for required for a supported authentication method. The evaluator shall configure a test endpoint to respond to an IKEv2 exchange with EAP-request, providing valid phase 1 handshake and valid TLS handshake, but computing the phase 2 shared key using standard (non-EAP) methods. The evaluator shall initiate an IPsec connection between the test endpoint and the TSF, and observe that the TSF aborts the connection.*

FCS_IPSEC_EXT.1 IPsec

FCS_IPSEC_EXT.1.1

The TSF shall implement IPsec as specified in [RFC 4301].

Application Note: [RFC 4301] calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a "traditional" SPD, etc. Regardless of the implementation details, there is a notion of a "rule" that a packet is "matched" against and a resulting action that takes place.

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the TOE can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface) but this is not required.

It is permissible for the TSF to receive configuration of the IPsec behavior from an environmental source. For example, The SPD is established and populated through an administrative interface or application implemented by the entity that establishes the IPsec connection, such as a VPN gateway or client application. This interface or application is outside the scope of this FP.

FCS_IPSEC_EXT.1.2

The TSF shall implement IPsec in [**selection**: *tunnel mode*, *transport mode*].

Application Note: If the TOE is used to connect to a VPN gateway for the purposes of establishing a secure connection to a private network, the ST author is expected to select tunnel mode. If the TOE uses IPsec to establish an end-to-end connection to another IPsec endpoint, the ST author is expected to select transport mode. If the TOE uses IPsec to establish a connection to a specific endpoint device for the purpose of secure remote administration, the ST author is expected to select transport mode.

FCS_IPSEC_EXT.1.3

The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by [RFC 4303] using the cryptographic algorithms [*AES-GCM-128 as specified in [RFC 4106], AES-GCM-256 as specified in [RFC 4106]*, [**selection**:

- *AES-CBC-128 with [**selection**: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] as specified in [RFC 3602],*
- *AES-CBC-256 with [**selection**: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] as specified in [RFC 3602],*
- *no other algorithms*

]].

Application Note: If this functionality is configurable, the TSF may be configured by a VPN gateway or by an Administrator of the TOE itself.

FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [**selection**:

- *IKEv1, using Main Mode for Phase I exchanges, as defined in [RFC 2407], [RFC 2408], [RFC 2409], [RFC 4109], [**selection**: [RFC 4304] for extended sequence numbers, no other RFCs for extended sequence numbers], [**selection**: [RFC 4868] for hash functions, no other RFCs for hash functions], and [**selection, choose one of**: support for XAUTH, no support for XAUTH],*
- *IKEv2 as defined in [RFC 7296] (with mandatory support for NAT traversal as specified in section 2.23), [RFC 8784], [RFC 8247], and [**selection**: [RFC 4868] for hash functions, no other RFCs for hash functions]*

].

FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [**selection**: *IKEv1*, *IKEv2*] protocol uses the cryptographic algorithms [*AES-CBC-128, AES-CBC-256 as specified in [RFC 6379] and* [**selection**: *AES-GCM-128 as specified in [RFC 5282], AES-GCM-256 as specified in [RFC 5282], no other algorithm*].

Application Note: If this functionality is configurable, the TSF may be configured by a VPN gateway or by an Administrator of the TOE itself.

FCS_IPSEC_EXT.1.7

The TSF shall ensure that [**selection**:

- *IKEv2 SA lifetimes can be configured by [**selection**: an Administrator, a VPN gateway] based on [**selection**: number of packets/number of bytes, length of time] ,*
- *IKEv1 SA lifetimes can be configured by [**selection**: an Administrator, a VPN gateway] based on [**selection**: number of packets/number of bytes, length of time] ,*
- *IKEv1 SA lifetimes are fixed based on [**selection**: number of packets/number of bytes, length of time]*

], and that if length of time is used, it shall include at least one option that is 24 hours or less for Phase 1 SAs and 8 hours or less for Phase 2 SAs.

Application Note: The ST author is afforded a selection based on the version of IKE in their implementation. There is a further selection within this selection that allows the ST author to specify which entity is responsible for “configuring” the life of the SA. An implementation that allows an administrator to configure the IPsec endpoint or a VPN gateway that pushes the SA lifetime down to the IPsec endpoint are both acceptable.

As far as SA lifetimes are concerned, the TOE can limit the lifetime based on the number of bytes transmitted, or the number of packets transmitted. Either packet-based or volume-based SA lifetimes are acceptable; the ST author makes the appropriate selection to indicate which type of lifetime limits are supported.

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5. The IKEv1 requirement can be accomplished either by providing Authorized Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE), or by "hard coding" the limits in the implementation. For IKEv2, there are no hard-coded limits, but in this case it is required that an administrator be able to configure the values. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE. It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

FCS_IPSEC_EXT.1.8

The TSF shall ensure that all IKE protocols implement DH Groups [*19 (256-bit Random ECP), 20 (384-bit Random ECP), and* [**selection**: *24 (2048-bit MODP with 256-bit POS), 15 (3072-bit MODP), 14 (2048-bit MODP), no other DH groups*]].

Application Note: The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS_CKM.1.

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignments in FCS_IPSEC_EXT.1.9 and FCS_IPSEC_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in 800-57 to determine the "bits of security" associated with the DH group. Each unique value is then used to fill in the assignment (for 1.9 they are doubled; for 1.10 they are inserted directly into the assignment). For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192. For FCS_IPSEC_EXT.1.9, then, the assignment would read "[224, 384]" and for FCS_IPSEC_EXT.1.10 it would read "[112, 192]" (although in this case the requirement should probably be refined so that it makes sense mathematically).

FCS_IPSEC_EXT.1.9

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**assignment**: *(one or more) number(s) of bits that is at least twice the "bits of security" value associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management: Part 1 - General*] bits.

FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[**assignment**: *(one or more) "bits of security" value(s) associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management: Part 1 - General*].

FCS_IPSEC_EXT.1.11

The TSF shall ensure that all IKE protocols perform peer authentication using [**selection**: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to [RFC 4945] and [**selection**: *Pre-shared Keys transmitted via EAP-TLS*, *Pre-shared Keys transmitted via EAP-TTLS with mutual authentication*, *Pre-shared Keys not transmitted via EAP*, *no other method*].

Application Note: At least one public-key-based Peer Authentication method is required in order to conform to this PP-Module; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, RFC 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).

If any selection with "pre-shared keys" is selected, the selection-based requirement FIA_PSK_EXT.1 must be claimed. FIA_PSK_EXT.1 includes the options for MFA solutions, it may be brought in via this selection or via the optional FPF_MFA_EXT.1 requirement.

When pre-shared keys are supported for IKE v2, 'Pre-shared Keys transmitted via EAP-TLS' and/or 'Pre-shared Keys transmitted via EAP-TTLS' is selected to indicate client verification using certificates in a mutually authenticated TLS

handshake, and verification of provided PSK protected under the TLS channel. When Pre-shared Keys are supported for IKE v1, the first selection is claimed to indicate one of the mechanisms for using PSK described in the RFC.

It is acceptable for different use cases to leverage different selections, if this is the case it shall be identified.

FCS_IPSEC_EXT.1.12

The TSF shall not establish an SA if the [[**selection**: *IP address*, *Fully Qualified Domain Name (FQDN)*, *user FQDN*, *Distinguished Name (DN)*] and [**selection**: *no other reference identifier type*, *[**assignment**: other supported reference identifier types]*]] contained in a certificate does not match the expected value(s) for the entity attempting to establish a connection.

Application Note: The TOE must support at least one of the following identifier types: IP address, Fully Qualified Domain Name (FQDN), user FQDN, or Distinguished Name (DN). In the future, the TOE will be required to support all of these identifier types. The TOE is expected to support as many IP address formats (IPv4 and IPv6) as IP versions supported by the TOE in general. The ST author may assign additional supported identifier types in the second selection.

FCS_IPSEC_EXT.1.13

The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

Application Note: At this time, only the comparison between the presented identifier in the peer's certificate and the peer's reference identifier is mandated by the testing below. However, in the future, this requirement will address two aspects of the peer certificate validation: 1) comparison of the peer's ID payload to the peer's certificate which are both presented identifiers, as required by RFC 4945 and 2) verification that the peer identified by the ID payload and the certificate is the peer expected by the TOE (per the reference identifier). At that time, the TOE will be required to demonstrate both aspects (i.e. that the TOE enforces that the peer's ID payload matches the peer's certificate which both match configured peer reference identifiers).

Excluding the DN identifier type (which is necessarily the Subject DN in the peer certificate), the TOE may support the identifier in either the Common Name or Subject Alternative Name (SAN) or both. If both are supported, the preferred logic is to compare the reference identifier to a presented SAN, and only if the peer's certificate does not contain a SAN, to fall back to a comparison against the Common Name. In the future, the TOE will be required to compare the reference identifier to the presented identifier in the SAN only, ignoring the Common Name.

The configuration of the peer reference identifier is addressed by FMT_SMF.1.1/VPN.

FCS_IPSEC_EXT.1.14

The [**selection**: *TSF*, *VPN gateway*] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [**selection**: *IKEv1 Phase 1*, *IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [**selection**: *IKEv1 Phase 2*, *IKEv2 CHILD_SA*] connection.

Application Note: If this functionality is configurable, the TSF may be configured by a VPN gateway or by an Administrator of the TOE itself

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE versions chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

Evaluation Activities

FCS_IPSEC_EXT.1
TSS
If the TOE boundary includes a general-purpose operating system or mobile device, the evaluator shall examine the TSS to ensure that it describes whether the IPsec functionality is

architecturally integrated with the platform itself or whether it is a separate executable that is bundled with the platform.

Guidance
The evaluator shall ensure that the operational guidance contains any appropriate information for ensuring that the IPsec implementation can be properly configured.

Note in this case that the implementation of the IPsec protocol must be enforced entirely within the TOE boundary; i.e. it is not permissible for a software application TOE to be a graphical front-end for IPsec functionality implemented totally or in part by the underlying OS platform. The behavior referenced here is for the possibility that the configuration of the IPsec connection is initiated from outside the TOE, which is permissible so long as the TSF is solely responsible for enforcing the configured behavior. However, it is allowable for the TSF to rely on low-level platform-provided networking functions to implement the SPD from the client (e.g., enforcement of packet routing decisions).

Tests
As a prerequisite for performing the Test EAs for the individual FCS_IPSEC_EXT.1 elements below, the evaluator must do the following:

The evaluator must create a test environment consisting of at least the components illustrated below. It is expected that the traffic generator will be used to construct network packets and will provide the evaluator with the ability manipulate fields in the ICMP, IPv4, IPv6, UDP, and TCP packet headers. The evaluator shall provide justification for any differences in the test environment.

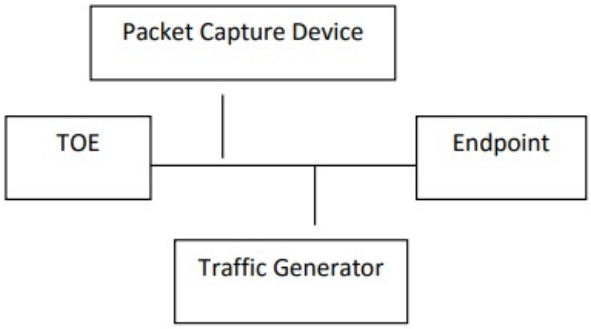

Figure 1: Test Environment

Note that the evaluator shall perform all tests using the TOE and a representative sample of platforms listed in the ST (for TOEs that claim to support multiple platforms).

FCS_IPSEC_EXT.1.1
TSS
The evaluator shall examine the TSS and determine that it describes how the IPsec functionality is implemented.

The evaluator shall ensure that the TSS identifies any platform functionality the TSF relies upon to support its IPsec implementation, if any (e.g. does it invoke cryptographic primitive functions from the platform's cryptographic library, enforcement of packet routing decisions by low-level network drivers).

In all cases, the evaluator shall also ensure that the TSS describes how the IPsec implementation interacts with the network stack of the platform(s) on which it can run (e.g., does the client insert itself within the stack via kernel mods, does the IPsec implementation simply invoke APIs to gain access to network services).

The evaluator shall ensure that the TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how the available rules and actions form the SPD using terms defined in RFC 4301 such as BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301. As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Guidance
The evaluator shall examine the operational guidance to verify that it describes how the SPD is created and configured. If there is an administrative interface to the IPsec implementation , then the guidance describes how the administrator specifies rules for processing a packet. The description includes all three cases - a rule that ensures packets are encrypted/decrypted, dropped, and allowing a packet to flow in plaintext. The evaluator shall determine that the

description in the operational guidance is consistent with the description in the TSS, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

If the IPsec implementation is configured by an external application, such as the VPN gateway, then the operational guidance should indicate this and provide a description of how IPsec is configured by the external application. The description should contain information as to how the SPD is established and set up in an unambiguous fashion. The description should also include what is configurable via the external application, how ordering of entries may be expressed, as well as the impacts that ordering of entries may have on the packet processing.

In either case, the evaluator ensures the description provided In the TSS is consistent with the capabilities and description provided in the operational guidance.

Tests
Depending on the implementation, the evaluator may be required to use a VPN gateway or some form of application to configure the client. For Test 2, the evaluator is required to choose an application that allows for the configuration of the full set of capabilities of the IPsec implementation . For example, if the IPsec implementation supports specification of wildcards, subnets, etc., it is unacceptable for the evaluator to specify only a single fully qualified IP addresses in the rule.

The evaluator shall perform the following tests:

- ***Test 1:*** *The evaluator shall configure an SPD that is capable of the following: dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the IPsec endpoint with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed through without modification, were encrypted by the IPsec implementation.*
- ***Test 2:*** *The evaluator shall devise several tests that cover a variety of scenarios for packet processing. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and operational guidance. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the operational guidance.*

FCS_IPSEC_EXT.1.2
TSS
The evaluator shall check the TSS to ensure it states that IPsec can be established to operate in tunnel mode or transport mode or both (as selected).

Guidance
The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection for each mode selected.

If both transport mode and tunnel mode are implemented, the evaluator shall review the operational guidance to determine how the use of a given mode is specified.

Tests
The evaluator shall perform the following tests based on the selections chosen:

- ***Test 1:*** *[conditional] If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures a VPN gateway to operate in tunnel mode. The evaluator configures the TOE and the VPN gateway to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN gateway peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using tunnel mode.*
- ***Test 2:*** *[conditional] If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures an IPsec peer to accept IPsec connections using transport mode. The evaluator configures the TOE and the endpoint device to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the remote endpoint. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.*
- ***Test 3:*** *[conditional] If both tunnel mode and transport mode are selected, the evaluator shall modify the testing for FCS_IPSEC_EXT.1 to include the supported mode for SPD PROTECT entries to show that they apply only to traffic that is transmitted or received using the indicated mode.*

FCS_IPSEC_EXT.1.3

TSS

The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no "rules" are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.

Guidance

The evaluator shall check that the operational guidance provides instructions on how to construct or acquire the SPD and uses the guidance to configure the TOE for the following test.

Tests

The evaluator shall perform the following test:

The evaluator shall configure the SPD such that it has entries that contain operations that DISCARD, PROTECT, and (if applicable) BYPASS network packets. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE-created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE's interfaces.

FCS_IPSEC_EXT.1.4

TSS

The evaluator shall examine the TSS to verify that the algorithms AES-GCM-128 and AES-GCM-256 are implemented. If the ST author has selected either AES-CBC-128 or AES-CBC-256 in the requirement, then the evaluator verifies the TSS describes these as well. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in the relevant iteration of FCS_COP.1 from the incorporating PP that applies to keyed-hash message authentication.

Guidance

The evaluator checks the operational guidance to ensure it provides instructions on how the TOE is configured to use the algorithms selected in this component and whether this is performed through direct configuration, defined during initial installation, or defined by acquiring configuration settings from an environmental component.

Tests

- ***Test 1:*** *The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the AES-GCM-128 and AES-GCM-256 algorithms, and attempt to establish a connection using ESP. If the ST Author has selected either AES-CBC-128 or AES-CBC-256, the TOE is configured to use those algorithms and the evaluator attempts to establish a connection using ESP for those algorithms selected.*

FCS_IPSEC_EXT.1.5

TSS

The evaluator shall examine the TSS to verify that IKEv1 or IKEv2 (or both) are implemented. If IKEv1 is implemented, the evaluator shall verify that the TSS indicates whether or not XAUTH is supported, and that aggressive mode is not used for IKEv1 Phase 1 exchanges (i.e. only main mode is used). It may be that these are configurable options.

Guidance

The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the test below. If XAUTH is implemented, the evaluator shall verify that the operational guidance provides instructions on how it is enabled or disabled.

If the TOE supports IKEv1, the evaluator shall verify that the operational guidance either asserts that only main mode is used for Phase 1 exchanges, or provides instructions for disabling aggressive mode.

Tests

- ***Test 1:*** *The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and [RFC 7296], section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed. If the TOE supports IKEv1 with or without XAUTH, the evaluator shall verify that this test can be successfully repeated with XAUTH enabled and disabled in the manner specified by the operational guidance. If the TOE supports only IKEv1 with XAUTH, the evaluator shall verify that connections not using XAUTH are unsuccessful. If the TOE supports only IKEv1 without XAUTH, the evaluator shall verify that connections using XAUTH are unsuccessful.*
- ***Test 2:*** *[conditional] If the TOE supports IKEv1, the evaluator shall perform any applicable operational guidance steps to disable the use of aggressive mode and then attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt*

should fail. The evaluator shall show that the TOE rejects a VPN gateway from initiating an IKEv1 Phase 1 connection in aggressive mode. The evaluator should then show that main mode exchanges are supported.

FCS_IPSEC_EXT.1.6
TSS
The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

Guidance
The evaluator checks the operational guidance to ensure it provides instructions on how the TOE is configured to use the algorithms selected in this component and whether this is performed through direct configuration, defined during initial installation, or defined by acquiring configuration settings from an environmental component.

Tests
The evaluator shall use the operational guidance to configure the TOE (or to configure the Operational Environment to have the TOE receive configuration) to perform the following test for each ciphersuite selected for each version of IKE selected:

The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 or IKEv2 payload and establish a connection with a peer device, which is configured to accept the payload encrypted only using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation. The evaluator will confirm that the connection is successful by confirming that data can be passed through the connection once it is established. For example, the evaluator may connect to a webpage on the remote network and verify that it can be reached.
FCS_IPSEC_EXT.1.7
TSS
There are no TSS EAs for this requirement.

Guidance
The evaluator shall check the operational guidance to ensure it provides instructions on how the TOE configures the values for SA lifetimes. In addition, the evaluator shall check that the guidance has the option for either the Administrator or VPN gateway to configure Phase 1 SAs if time-based limits are supported. Currently there are no values mandated for the number of packets or number of bytes, the evaluator shall simply check the operational guidance to ensure that this can be configured if selected in the requirement.

Tests
When testing this functionality, the evaluator needs to ensure that both IPsec endpoints are configured appropriately. From the RFC: "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

- ***Test 1:** The evaluator shall configure a maximum lifetime in terms of the number of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed number of packets (or bytes) through this SA is exceeded, the connection is closed.*
- ***Test 2:** The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.*
- ***Test 3:** The evaluator shall perform a test similar to Test 2 for Phase 2 SAs, except that the lifetime will be 8 hours or less instead of 24 hours or less.*
- ***Test 4:** If a fixed limit for IKEv1 SAs is supported, the evaluator shall establish an SA and observe that the connection is closed after the fixed traffic or time value is reached.*

FCS_IPSEC_EXT.1.8
TSS
The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks

to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Guidance
There are no AGD EAs for this requirement.

Tests
The evaluator shall perform the following test:

- ***Test 1:*** *For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.*

FCS_IPSEC_EXT.1.9
TSS
The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.9) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this FP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

Guidance
There are no AGD EAs for this requirement.

Tests
There are no test EAs for this requirement.
FCS_IPSEC_EXT.1.10
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.9.
FCS_IPSEC_EXT.1.11
TSS
The evaluator ensures that the TSS identifies RSA or ECDSA or both as being used to perform peer authentication.

If any selection with pre-shared keys is chosen in the selection, the evaluator shall check to ensure that the TSS describes how those selections work in conjunction with authentication of IPsec connections.

The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include whether the certificate presented identifier is compared to the ID payload presented identifier, which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN) and, if multiple fields are supported, the logical order comparison. If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate.

Guidance
If any selection with "Pre-shared Keys" is selected, the evaluator shall check that the operational guidance describes any configuration necessary to enable any selected authentication mechanisms.

If any method other than no other method is selected, the evaluator shall check that the operational guidance describes any configuration necessary to enable any selected authentication mechanisms.

The evaluator ensures the operational guidance describes how to set up the TOE to use the cryptographic algorithms RSA or ECDSA, as selected.

In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operational guidance also describes how to configure the TOE to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE as a trusted CA.

The evaluator shall also ensure that the operational guidance includes the configuration of the reference identifiers for the peer.

Tests
For efficiency's sake, the testing that is performed here has been combined with the testing for ***FIA_X509_EXT.2*** *and* ***FIA_X509_EXT.3*** *(for IPsec connections and depending on the Base-PP),* ***FCS_IPSEC_EXT.1.12****, and* ***FCS_IPSEC_EXT.1.13****. The following tests shall be repeated for each peer authentication protocol selected in the* ***FCS_IPSEC_EXT.1.11*** *selection above:*

- ***Test 1:*** *The evaluator shall have the TOE generate a public-private key pair, and submit a CSR (Certificate Signing Request) to a CA (trusted by both the TOE and the peer VPN used to establish a connection) for its signature. The values for the DN (Common Name, Organization, Organizational Unit, and Country) will also be passed in the request. Alternatively, the evaluator may import to the TOE a previously generated private key and corresponding certificate.*
- ***Test 2:*** *The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.*
- ***Test 3:*** *The evaluator shall test that the TOE can properly handle revoked certificates –*

conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. For this current version of the PP-Module, the evaluator has to only test one up in the trust chain (future drafts may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the SA is established. The evaluator then attempts the test with a certificate that will be revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the TOE will not establish an SA

- ***Test 4:*** *[conditional] For each selection made, the evaluator shall verify factors are required, as indicated in the operational guidance, to establish an IPsec connection with the server.*
  For each supported identifier type (excluding DNs), the evaluator shall repeat the following tests:
- ***Test 5:*** *For each field of the certificate supported for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds.*
- ***Test 6:*** *For each field of the certificate support for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to not match the field in the peer's presented certificate and shall verify that the IKE authentication fails.*
  The following tests are conditional:
- ***Test 7:*** *[conditional] If, according to the TSS, the TOE supports both Common Name and SAN certificate fields and uses the preferred logic outlined in the Application Note, the tests above with the Common Name field shall be performed using peer certificates with no SAN extension. Additionally, the evaluator shall configure the peer's reference identifier on the TOE to not match the SAN in the peer's presented certificate but to match the Common Name in the peer's presented certificate, and verify that the IKE authentication fails.*
- ***Test 8:*** *[conditional] If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds. To demonstrate a bit-wise comparison of the DN, the evaluator shall change a single bit in the DN (preferably, in an Object Identifier (OID) in the DN) and verify that the IKE authentication fails.* ***To demonstrate a comparison of DN values, the evaluator shall change any one of the four DN values and verify that the IKE authentication fails.***
- ***Test 9:*** *[conditional] If the TOE supports both IPv4 and IPv6 and supports IP address identifier types, the evaluator must repeat test 1 and 2 with both IPv4 address identifiers and IPv6 identifiers. Additionally, the evaluator shall verify that the TOE verifies that the IP header matches the identifiers by setting the presented identifiers and the reference identifier with the same IP address that differs from the actual IP address of the peer in the IP headers and verifying that the IKE authentication fails.*
- ***Test 10:*** *[conditional] If, according to the TSS, the TOE performs comparisons between the peer's ID payload and the peer's certificate, the evaluator shall repeat the following test for each combination of supported identifier types and supported certificate fields (as above). The evaluator shall configure the peer to present a different ID payload than the field in the peer's presented certificate and verify that the TOE fails to authenticate the IKE peer.*

FCS_IPSEC_EXT.1.12
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.11.
FCS_IPSEC_EXT.1.13
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.11.
FCS_IPSEC_EXT.1.14
TSS
The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA negotiation that is being protected.

Guidance
There are no AGD EAs for this requirement.

Tests
The evaluator follows the guidance to configure the TOE to perform the following tests for each version of IKE supported:

- ***Test 1:*** *The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.*
- ***Test 2:*** *The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with greater strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.*
- ***Test 3:*** *The evaluator shall attempt to establish an IKE SA using an algorithm that is not*

one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

- ***Test 4:*** *The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.*

3.3 Identification and Authentication (FIA)

The TOE may support pre-shared keys for use in establishing an IPsec connection. PSK in the context of this document refer to generated values, memorized values subject to conditioning, one time passwords, and combinations of the above as described in FIA_PSK_EXT.1.2.

FIA_HOTP_EXT.1 HMAC-Based One-Time Password Pre-Shared Keys

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.4.2.

FIA_HOTP_EXT.1.1

The TSF shall support HMAC-Based One-Time Password authentication (HOTP) in accordance with [RFC 4226] to authenticate the user before establishing an IPsec connection.

FIA_HOTP_EXT.1.2

The TSF shall generate a HOTP seed of [**selection**: *128*, *256*] bits in accordance with FCS_RBG_EXT.1.

FIA_HOTP_EXT.1.3

The TSF shall generate a new HOTP seed value for each IPsec connection.

FIA_HOTP_EXT.1.4

The TSF shall utilize [**selection**: *HMAC-SHA-1*, *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] to derive a HOTP hash from the HOTP seed and counter.

Application Note: The assignment must be consistent with the key sizes supported by the relevant iteration of FCS_COP.1.

FIA_HOTP_EXT.1.5

The TSF shall truncate the HOTP hash per FIA_HOTP_EXT.1.4 to create a HOTP of [**selection, choose one of**:

- *administrator configurable character length of at least 6*,
- *preset character length of [**selection, choose one of**: 6, 7, 8, 9, 10]*

].

Application Note: The ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

FIA_HOTP_EXT.1.6

The TSF shall [**selection**:

- *throttle invalid requests to [**selection**: administrator configurable value, [**assignment**: value less than 10]] per minute* ,
- *lock the associated account after [**selection**: administrator configurable value, [**assignment**: value less than 10]] failed attempts until [**selection**: an administrator unlocks the account, a configurable time period]*

].

Application Note: The ST author may select throttle requests, account lockout, or both.

FIA_HOTP_EXT.1.7

The TSF shall not verify HOTP attempts outside of the counter look ahead window of [**selection**: *a configurable value*, *[**assignment**: a value less than or equal to 3]*] for resynchronization.

FIA_HOTP_EXT.1.8

The TSF shall increment the counter after each successful authentication.

Application Note: The HOTP seed and all derived values are considered secret keys for purposes of protection. This requirement must be claimed if "*verify the HOTP*" is selected in FIA_PSK_EXT.4.2.

Evaluation Activities

FIA_HOTP_EXT.1
TSS
The evaluator shall confirm the TSS describes how the TOE complies with [RFC 4226].

The evaluator shall confirm the TSS describes how the HOTP seed is generated and ensure it aligns with FCS_RBG_EXT.1.

The evaluator shall confirm the TSS describes how the HOTP seed is protected and ensure it aligns with the storage requirements of the base PP.

The evaluator shall confirm the TSS describes how a new HOTP seed is assigned for each client and how each client is uniquely identified.

The evaluator shall confirm the TSS describes how the HOTP seed is conditioned into a HOTP hash and verify it matches the selection in FIA_HOTP_EXT.1.4.

The evaluator shall confirm the TSS describes how the HOTP hash is truncated and verify it matches the selection in FIA_HOTP_EXT.1.5.

The evaluator shall confirm the TSS describes how the TOE handles multiple incoming invalid requests and verify it provides anti-hammer mechanism that match the selections FIA_HOTP_EXT.1.6.

The evaluator shall confirm the TSS describes how the TOE handles resynchronization and how it rejects attempts outside of the look-ahead window selected in FIA_TOTP_EXT.1.7.

The evaluator shall confirm the TSS describes how the TOE how the counter is incremented after each successful authentication.

Guidance
The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_HOTP_EXT.1 requirements.
Tests
The evaluator shall configure the TOE to use a supported HOTP factor then:

- ***Test 1:*** *Attempt to establish a connection using a factor from a different client, the test passes if the client fails to connect.*
- ***Test 2:*** *Attempt multiple connections outside the boundary set in FIA_HOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- ***Test 3:*** *Attempt to use a HOTP that is outside of the value allowed with for resynchronization. The test passes if the client fails to connect.*
- ***Test 4:*** *Attempt to connect with a valid HOTP, disconnect and attempt to authenticate again with the same HOTP value. The test passes if the client connects the first time and fails to connect the second time. If the HOTP generated is duplicated the test may be repeated.*

FIA_PSK_EXT.1 Pre-Shared Key Composition

This is a selection-based component. Its inclusion depends upon selection from FCS_IPSEC_EXT.1.11, FCS_IPSEC_EXT.1.11, FCS_IPSEC_EXT.1.11.

FIA_PSK_EXT.1.1

The TSF shall be capable of using pre-shared keys for establishing IPsec connections.

FIA_PSK_EXT.1.2

The TSF shall be able to accept the following as pre-shared keys: [**selection**: *Generated bit-based, Password-based, HMAC based one time password, Time based one time password, Combination of a generated bit-based and HMAC-based one-time password, Combination of a generated bit-based and time-based one-time password, Combination of a password-based and HMAC-based one-time password, Combination of a password-based and time-based one-time password*] keys.

Application Note: If any selection including "*generated bit-based*" keys is selected, then FIA_PSK_EXT.2 must be claimed.

If any selection including "*password-based*" keys is selected then FIA_PSK_EXT.3 must be claimed.

If any selection including "*HMAC-based one-time password*" keys is selected then FIA_PSK_EXT.4 must be claimed.

If any selection including "*time-based one-time password*" is selected then FIA_PSK_EXT.5 must be claimed.

This requirement must be claimed if "*Pre-shared keys*" is selected in FCS_IPSEC_EXT.1.11.

Evaluation Activities

FIA_PSK_EXT.1
TSS
The evaluator shall examine the TSS to ensure that it identifies that IPsec connections may use pre-shared keys. The evaluator shall confirm that the TSS states which pre-shared key selections are supported.
Guidance
The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required.
Tests
The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE).

- ***Test 1:*** *For each mechanism selected in FIA_PSK_EXT.1.2 the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection.*

FIA_PSK_EXT.2 Generated Pre-Shared Keys

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.

FIA_PSK_EXT.2.1

The TSF shall be able to [**selection**:

- *accept externally generated*,
- *generate [**selection**: 128, 256] bit-based pre-shared keys via FCS_RBG_EXT.1.*

]

Application Note: Generated PSKs are expected to be shared between components via an out of band mechanism. This requirement must be claimed if any selection in FIA_PSK_EXT.1.2 includes "*generated bit-based*" keys.

Evaluation Activities

FIA_PSK_EXT.2
TSS
If "generate" is selected the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1 and the output matches the size selected in FIA_PSK_EXT.2.1.
Guidance
The evaluator shall confirm the operational guidance contains instructions for entering generated pre-shared keys.
Tests

- ***Test 1:*** *[conditional] If "generate" was selected the evaluator shall generate a pre-shared key and confirm the output matches the size selected in FIA_PSK_EXT.2.1.*

FIA_PSK_EXT.3 Password-Based Pre-Shared Keys

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.

FIA_PSK_EXT.3.1

The TSF shall support a PSK of up to [**assignment**: *positive integer of 64 or more*] characters.

Application Note: The ST author assigns the maximum size of the PSK it supports; it must support at least 64 characters or a length defined by the platform.

FIA_PSK_EXT.3.2

The TSF shall allow PSKs to be composed of any combination of upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [**selection**: *[**assignment**: other supported special characters], no other characters*]

Application Note: The ST author assigns any other supported characters; if there are no other supported characters, then "*no other characters*" should be selected.

FIA_PSK_EXT.3.3

The TSF shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [**selection**: *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*], with [**assignment**: *positive integer of 4096 or more*] iterations], and output cryptographic key sizes [**selection**: *128*, *256*] that meet the following: [NIST SP 800-132 Part 1].

Application Note: The ST author selects the parameters based on the PBKDF used by the TSF.

FIA_PSK_EXT.3.4

The TSF shall not accept PSKs less than [**selection**: *a value settable by the administrator*, *[**assignment**: minimum PSK length accepted by the TOE, must be >= 6]*] and greater than the maximum PSK length defined in FIA_PSK_EXT.3.1.

Application Note: If the minimum length is settable, then ST author chooses "*a value settable by the administrator*". If the minimum length is not settable, the ST author fills in the assignment with the minimum length the PSK must be. This requirement is to ensure bounds work properly.

FIA_PSK_EXT.3.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 and with entropy corresponding to the key size selected for PBKDF in FIA_PSK_EXT.3.3.

FIA_PSK_EXT.3.6

The TSF shall require the PSK to be entered before every initiated connection.

FIA_PSK_EXT.3.7

The TSF shall [**selection**: *provide a password strength meter*, *check the password against a blacklist*, *perform no action to assist the user in choosing a strong password*].

Application Note: For FIA_PSK_EXT.3.7, the ST author may select one, both, or neither of the functions in alignment with [NIST SP 800-63B].

This requirement is selection dependent on FIA_PSK_EXT.1.

Evaluation Activities

FIA_PSK_EXT.3
TSS
The evaluator shall examine the TSS to ensure it describes the process by which the bit-based pre-shared keys used. If generated is selected the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

Support for length: The evaluators shall check to ensure that the TSS describes the allowable ranges for PSK lengths, and that at least 64 characters or a length defined by the platform may be specified by the user.

Support for character set: The evaluator shall check to ensure that the TSS describes the allowable character set and that it contains the characters listed in the SFR.

Support for PBKDF: The evaluator shall examine the TSS to ensure that the use of PBKDF2 is described and that the key sizes match that described by the ST author.

The evaluator shall check that the TSS describes the method by which the PSK is first encoded and then fed to the hash algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself.

For the NIST SP 800-132-based conditioning of the PSK, the required evaluation activities will be performed when doing the evaluation activities for the appropriate requirements (e.g., FCS_COP.1/KeyedHash).

The evaluator shall confirm that the minimum length is described.

The ST author shall provide a description in the TSS regarding the salt generation. The evaluator shall confirm that the salt is generated using an RBG described in FCS_RBG_EXT.1.

[conditional] If "password strength meter" or "password blacklist" is selected, the evaluator shall examine the TSS to ensure any password checking functionality provided by the TSF is described and contains details on how the function operates.

Guidance
The evaluator shall confirm the operational guidance contains instructions for entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall confirm that any configuration management functions related to pre-shared keys that are performed by the TOE are specified in the operational guidance.

The guidance must specify the allowable characters for pre-shared keys, and that list must include, at minimum, the same items contained in FIA_PSK_EXT.3.2.

The evaluator shall confirm the operational guidance contains any necessary instructions for enabling and configurating password checking functionality.

Tests
Support for Password/Passphrase characteristics: In addition to the analysis above, the evaluator shall also perform the following tests on a TOE configured according to the Operational Guidance:

- ***Test 1:** The evaluator shall compose a pre-shared key of at least 64 characters that contains a combination of the allowed characters in accordance with the FIA_PSK_EXT.1.3 and verify that a successful protocol negotiation can be performed with the key.*
- ***Test 2:** [conditional] If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length and invalid lengths that are below the minimum length, above the maximum length, null length, empty length, or zero length. The minimum test should be successful, and the invalid lengths must be rejected by the TOE.*
- ***Test 3:** [conditional] If the TOE initiates connections, initiate and establish a remote connection, disconnect from the connection, verify that the PSK is required when initiating the connection a second time.*
- ***Test 4:** [conditional] If the TOE supports a password meter, the evaluator shall enter a password and verify the password checker responds per the description in the TSS.*
- ***Test 5:** [conditional] If the TOE supports a password blacklist, the evaluator shall enter a blacklisted password and verify that the password is rejected or flagged as such.*

FIA_PSK_EXT.4 HMAC-Based One-Time Password Pre-shared Keys Support

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.

FIA_PSK_EXT.4.1

The TSF shall accept and send a HOTP while initiating an IPsec connection.

FIA_PSK_EXT.4.2

The TSF shall [**selection, choose one of**: *verify the HOTP*, *verify the HOTP via an external authentication server*] before establishing an incoming connection.

Application Note: If "*verify the HOTP*" is selected, then FIA_HOTP_EXT.1 must be included.

This requirement must be claimed if "*HMAC-based one-time password*" is included in a selection in FIA_PSK_EXT.1.2.

Evaluation Activities

FIA_PSK_EXT.4
TSS
The evaluator shall verify the TSS describes how the HOTP is input into the client and how that value is sent to the server.

The evaluator shall verify the TSS describes how the HOTP is accepted from an incoming connection and how that value is verified, either by the TOE or by an external authentication server.

Guidance
The evaluator shall verify the operational guidance contains any configuration necessary to enable HOTP.

Tests

- ***Test 1:** The evaluator shall configure the TOE to use a supported HOTP factor, then*

attempt to establish a connection using that factor.
- ***Test 2:*** *The evaluator shall verify the client prompts the user for the HOTP before initiating the connection.*
- ***Test 3:*** *The evaluator shall verify the server validates the HOTP or receives confirmation from an authentication server before establishing the channel.*

FIA_PSK_EXT.5 Time Based One Time Password Pre-shared Keys Support

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2, FIA_PSK_EXT.1.2.

FIA_PSK_EXT.5.1

The TSF shall accept and send a TOTP while initiating an IPsec connection.

FIA_PSK_EXT.5.2

The TSF shall [**selection, choose one of**: *verify the TOTP*, *verify the TOTP via an external authentication server*] before establishing an incoming connection.

Application Note: If "*verify the TOTP*" is selected then FIA_TOTP_EXT.1 must be claimed.

This requirement must be claimed if "*Time-based one-time password*" is included in a selection in FIA_PSK_EXT.1.2.

Evaluation Activities

FIA_PSK_EXT.5
TSS
The evaluator shall verify the TSS describes how the TOTP is input into the client and how that value is sent to the server.

The evaluator shall verify the TSS describes how the TOTP is accepted from an incoming connection and how that value is verified, either by the TOE or by an external authentication server.

Guidance
The evaluator shall verify the operational guidance contains any configuration necessary to enable TOTP.
Tests
- ***Test 1:*** *The evaluator shall configure the TOE to use a supported TOTP factor, then attempt to establish a connection using that factor.*
- ***Test 2:*** *The evaluator shall verify the client prompts the user for the TOTP before initiating the connection.*
- ***Test 3:*** *The evaluator shall verify the server validates the TOTP or receives confirmation from an authentication server before establishing the channel.*

FIA_TOTP_EXT.1 Time-Based One-Time Password Pre-Shared Keys

This is a selection-based component. Its inclusion depends upon selection from FIA_PSK_EXT.5.2.

FIA_TOTP_EXT.1.1

The TSF shall support Time-Based One-Time Password (TOTP) authentication in accordance with [RFC 6238] to authenticate the user before establishing an IPsec connection.

FIA_TOTP_EXT.1.2

The TSF shall generate a TOTP seed according to FCS_RBG_EXT.1 of [**selection**: *128*, *256*] bits.

FIA_TOTP_EXT.1.3

The TSF shall generate a new TOTP seed for each IPsec connection.

FIA_TOTP_EXT.1.4

The TSF shall utilize [**selection**: *HMAC-SHA-1*, *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] to derive a TOTP hash from the TOTP seed and current time provided by NTP.

Application Note: The selection must be consistent with the key sizes

permitted by the relevant iteraton of FCS_COP.1.

FIA_TOTP_EXT.1.5

The TSF shall truncate the TOTP hash per FIA_TOTP_EXT.1.4 to create a TOTP of [**selection**:

- *administrator configurable character length of at least 6,*
- *preset character length of [**selection, choose one of**: 6, 7, 8, 9, 10]*

].

Application Note: The ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

FIA_TOTP_EXT.1.6

The TSF shall [**selection**:

- *throttle invalid requests to [**selection**: administrator configurable value, [**assignment**: value less than 10]] per minute,*
- *lock the associated account after [**selection**: administrator configurable value, [**assignment**: value less than 10]] failed attempts until [**selection**: an administrator unlocks the account, a configurable time period]*

].

Application Note: The ST may select throttle requests, account lockout, or both.

FIA_TOTP_EXT.1.7

The TSF shall set a time-step size of [**selection**: *a configurable value, [**assignment**: a value less than or equal to 30]*] seconds.

FIA_TOTP_EXT.1.8

The TSF shall not validate a drift of more than [**selection**: *a configurable value, [**assignment**: a value less than or equal to 3]*] time-steps.

FIA_TOTP_EXT.1.9

The TSF shall [**selection, choose one of**: *allow resynchronization by recording time drift within the limit of FIA_TOTP_EXT.2.8, not permit resynchronization*].

Application Note: The TOTP seed and all derived values are considered secret keys for purposes of protection. This requirement must be claimed if "verify the TOTP" is selected in FCS_PSK_EXT.5.2.

Evaluation Activities

FIA_TOTP_EXT.1
TSS
The evaluator shall confirm the TSS describes how the TOE complies with [RFC 6238]

. The evaluator shall confirm the TSS describes how the TOTP seed is generated and ensure it aligns with FCS_RBG_EXT.1.

The evaluator shall confirm the TSS describes how the TOTP seed is protected and ensure it aligns with the storage requirements of the base PP.

The evaluator shall confirm the TSS describes how a new TOTP seed is assigned for each client and how each client is uniquely identified.

The evaluator shall confirm the TSS describes how the TOTP seed is conditioned into a TOTP hash and verify it matches the selection in FIA_TOTP_EXT.1.4.

The evaluator shall confirm the TSS describes how the TOTP hash is truncated and verify it matches the selection in FIA_TOTP_EXT.1.5.

The evaluator shall confirm the TSS describes how the TOE handles multiple incoming requests and verify it provides anti-hammer mechanism that matches the selections FIA_TOTP_EXT.2.6.

The evaluator shall confirm the TSS describes how the TOE sets a time-step value and verify it matches the selections in the ST.

The evaluator shall confirm the TSS describes how the TOE handles drift and resynchronization and verify it matches the selections. The evaluator shall ensure the TSS describes how time is kept and drift is calculated. If drift is recorded the evaluator shall ensure the TSS how this is done.

Guidance
The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_TOTP_EXT.1 requirements.
Tests
The evaluator shall configure the TOE to use a supported TOTP factor then:

- ***Test 1:*** *Attempt to establish a connection using a factor from a different client, the test passes if the client fails to connect.*
- ***Test 2:*** *Attempt multiple connections outside the boundary set in FIA_TOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- ***Test 3:*** *Attempt to use a TOTP that is outside of the value allowed with for resynchronization. The test passes if the client fails to connect.*
- ***Test 4:*** *Attempt to connect with a valid TOTP, disconnect and attempt to authenticate again with the same TOTP. The test passes if the client connects the first time and fails to connect the second time. If the TOTP generated is duplicated the test may be repeated.*

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [**selection, choose one of**: *invoke platform-provided functionality*, *implement functionality*] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [**selection**: *OCSP as specified in RFC 6960*, *CRL as specified in RFC 5280 Section 6.3*, *CRL as specified in RFC 8603*, *an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066*, *OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961*].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

Application Note: FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for HTTPS, TLS, and DTLS; this use requires that the extendedKeyUsage rules are verified. If OCSP is not supported the EKU provision for checking the OCSP Signing purpose is met by default.

This requirement is included if the protocol(s) selected in FTP_DIT_EXT.1.1 require the use of certificates. If the TOE implements TLS as a HTTPS/TLS server with no mutual authentication, this requirement is not applicable.

OCSP stapling and OCSP multi-stapling only support TLS server certificate validation. If other certificate types are validated, either OCSP or CRL should be claimed.

Regardless of the selection of "*implement functionality or invoke platform-provided functionality*," the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

FIA_X509_EXT.1.2

The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note: This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

Evaluation Activities

FIA_X509_EXT.1.1
TSS
The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Guidance
None.

Tests
The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

- ***Test 1:*** *The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:*
  - *by establishing a certificate path in which one of the issuing certificates is not a CA certificate,*
  - *by omitting the basicConstraints field in one of the issuing certificates,*
  - *by setting the basicConstraints field in an issuing certificate to have CA=False,*
  - *by omitting the CA signing bit of the key usage field in an issuing certificate, and*
  - *by setting the path length field of a valid CA field to a value strictly less than the certificate path.*

  The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.
- ***Test 2:*** *The evaluator shall demonstrate that validating an expired certificate results in the function failing.*
- ***Test 3:*** *The evaluator shall test that the TOE can properly handle revoked certificates-"conditional on whether CRL, OCSP, OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:*
  - *The evaluator shall test revocation of the node certificate.*
  - *The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted.*
  - *The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.*
- ***Test 4:*** *If any OCSP option is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.*
- ***Test 5:*** *The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)*
- ***Test 6:*** *The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*
- ***Test 7:*** *The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*
- ***Test 8:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.*
- ***Test 9:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall replace the intermediate certificate in the certificate chain for Test 9 with a*

modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 9, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

FIA_X509_EXT.1.2
TSS
None.

Guidance
None.

Tests
The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

- ***Test 1:*** *The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store.*
- ***Test 2:*** *The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store.*

Appendix A - Implementation-Dependent Requirements

Implementation-Dependent Requirements are dependent on the TOE implementing a particular function. If the TOE fulfills any of these requirements, the vendor must either add the related SFR or disable the functionality for the evaluated configuration.

Appendix B - Use Case Templates

B.1 IPsec Endpoint

The configuration for *[USE CASE 1] IPsec Endpoint* modifies the base requirements as follows:

- Include FCS_IPSEC_EXT.1 in the ST

B.2 EAP

The configuration for *[USE CASE 2] EAP* modifies the base requirements as follows:

B.3 Pre-Shared Keys

The configuration for *[USE CASE 3] Pre-Shared Keys* modifies the base requirements as follows:

- Include FIA_PSK_EXT.1 in the ST

Appendix C - Acronyms

Acronym	Meaning
Base-PP	Base Protection Profile
CA	Certificate Authority
CC	Common Criteria
CEM	Common Evaluation Methodology
CSR	Certificate Signing Request
DN	Distinguished Name
EAP	Extensible Authentication Protocol
ECP	Eliptic Curve group modulo a Prime
EP	Extended Package
ESN	Extended Sequence Number
ESP	Encapsulating Security Payload
FP	Functional Package
FQDN	Fully Qualified Domain Name
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
OCSP	Online Certificate Status Protocol
OE	Operational Environment
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module
PSK	Pre-Shared Key
SA	Security Association
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SPD	Security Policy Database
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
UDP	User Datagram Protocol
VPN	Virtual Private Network
XAUTH	Extended Authentication
cPP	Collaborative Protection Profile

Appendix D - Bibliography

Identifier	Title
[Functional Package for TLS]	Functional Package for Transport Layer Security (TLS)
[NIST SP 800-132 Part 1]	Recommendation for Password-Based Key Derivation Part 1: Storage Applications
[NIST SP 800-57 Part 1 Rev. 5]	Recommendation for Key Management: Part 1 - General
[NIST SP 800-63B]	Digital Identity Guidelines: Authentication and Lifecycle Management
[RFC 2407]	The Internet IP Security Domain of Interpretation for ISAKMP
[RFC 2408]	Internet Security Association and Key Management Protocol (ISAKMP)
[RFC 2409]	The Internet Key Exchange (IKE)
[RFC 3602]	The AES-CBC Cipher Algorithm and Its Use with IPsec
[RFC 4106]	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
[RFC 4109]	Algorithms for Internet Key Exchange version 1 (IKEv1)
[RFC 4226]	HOTP: An HMAC-Based One-Time Password Algorithm
[RFC 4301]	Security Architecture for the Internet Protocol
[RFC 4303]	IP Encapsulating Security Payload (ESP)
[RFC 4304]	Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)
[RFC 4868]	Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec
[RFC 4945]	The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
[RFC 5216]	The EAP-TLS Authentication Protocol
[RFC 5281]	Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)
[RFC 5282]	Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol
[RFC 6238]	TOTP: Time-Based One-Time Password Algorithm
[RFC 6379]	Suite B Cryptographic Suites for IPsec
[RFC 7296]	Internet Key Exchange Protocol Version 2 (IKEv2)
[RFC 8247]	Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)
[RFC 8784]	Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security
[RFC 8996]	Deprecating TLS 1.0 and TLS 1.1
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.