# Protection Profile for IPsec Virtual Private Network (VPN) Clients

Information Assurance Directorate

12 April 2013

Version 1.3

# Table of Contents

# List of Tables

# List of Figures

# Revision History

| Version | Date | Description |
|---|---|---|
| 1.0 | December 2011 | Initial release |
| 1.1 | December 2012 | Minor update. To make cryptographic requirements consistent with the VPN Gateway Extended Package. |
| 1.2 | January 2013 | Updated **FCS_COP.1.1(2)** to make consistent with the VPN Gateway Extended Package. |
| 1.3 | April 2013 | Updated X.509 requirements to specify the certificate path validation algorithm must ensure a basicConstraints field is present and the cA flag set to TRUE as a condition that must be met for a certificate to be considered a CA certificate. |

# 1 Introduction to the PP

1 This Protection Profile (PP) supports procurements of commercial off-the-shelf (COTS) IPsec Virtual Private Network (VPN) Clients to provide secure tunnels to authenticated remote endpoints or gateways. This PP details the policies, assumptions, threats, security objectives, security functional requirements, and security assurance requirements for the VPN and its supporting environment.

2 The primary intent is to clearly communicate to developers our understanding of the Security Functional Requirements needed to counter the threats that are being addressed by the VPN Client. The description in the TOE Summary Specification (TSS) of the Security Target (ST) is expected to document the architecture of the product (Target of Evaluation) and the mechanisms used to ensure that critical security transactions are correctly implemented.

## 1.1 PP Overview of the TOE

3 This document specifies Security Functional Requirements (SFRs) for a VPN Client. A VPN provides a protected transmission of private data between VPN Clients and VPN Gateways. The TOE defined by this PP is the VPN Client, a component executing on a remote access client. The VPN Client is intended to be located outside or inside of a private network, and provides a secure tunnel to a VPN Gateway. The tunnel provides confidentiality, integrity, and data authentication for information that travels across the public network. All VPN clients that comply with this document will support IPsec.

### 1.1.1 Usage and major security features of TOE

4 A VPN Client allows remote users to use client computers to establish an encrypted IPsec tunnel across an unprotected public network to a private network (see Figure 1). The TOE sits between the public network and entities (software, users, etc.) that reside on the VPN Client's underlying OS. IP packets crossing from the private network to the public network will be encrypted if their destination is a remote access VPN Client supporting the same VPN policy as the source network. The VPN Client protects the data between itself and a VPN Gateway, providing confidentiality, integrity, and protection of data in transit, even though it traverses a public network.

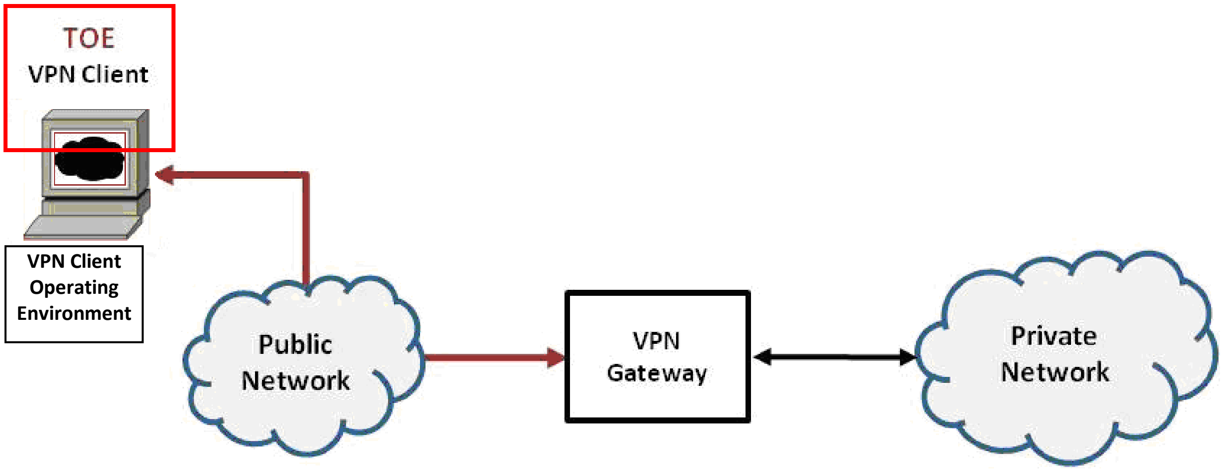


Figure 1: VPN Client

5 The focus of the Security Functional Requirements in this PP is on the following fundamental aspects of a VPN Client:

- Authentication of the VPN Gateway;
- Cryptographic protection of data in transit; and
- Implementation of services.

6 A VPN client can establish VPN connectivity with another VPN endpoint client or a VPN Gateway (that is the "remote" endpoint in the VPN communication). VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity. Authentication of a VPN Gateway is performed as part of the Internet Key Exchange (IKE) negotiation. The IKE negotiation uses a pre-existing public key infrastructure for authentication and can optionally use a pre-shared key. When IKE completes, an IPsec tunnel secured with Encapsulating Security Payload (ESP) is established.

7 It is assumed that the VPN Client is implemented properly and contains no critical design mistakes. The VPN Client relies on the IT environment for its proper execution as well as the following client machine protection mechanisms: audit review, audit storage, identification and authentication, security management, and session management. The vendor is required to provide configuration guidance (AGD_PRE, AGD_OPE) to correctly install and administer the client machine and the TOE for every operational environment supported.

### 1.1.2 Cryptography

8 The IPsec VPN Client is expected to encrypt all information that flows between itself and its VPN Gateway. The VPN Client serves as an endpoint for an IPsec VPN tunnel and performs a number of cryptographic functions related to establishing and maintaining the tunnel. If the cryptography used to authenticate, generate keys, and encrypt information is sufficiently robust and the implementation has no critical design mistakes, an adversary will be unable to exhaust the encryption key space to obtain the data. Compliance with IPsec standards, use of a properly seeded Random Bit Generator (RBG), and secure authentication factors will ensure that access to the transmitted information cannot be obtained with less work than a full exhaust of the key space. Any plaintext secret and private keys or other cryptographic security parameters will be zeroized when no longer in use to prevent disclosure of security critical data.

### 1.1.3 TOE Administration and the IT Environment

9 The TOE supporting environment is significant. The TOE in almost all cases will be a purely a software solution executing on a general purpose operating system. As such, the TOE must rely heavily on the TOE Operational Environment (system hardware, firmware, and operating system) for its execution domain and its proper usage. The vendor is expected to provide sufficient installation and configuration instructions to identify an Operational Environment with the necessary features and to provide instructions for how to configure it correctly.

The TOE requires that certain management activities (defined in the requirements) be performed by a subset of the authorized users of the TOE. This PP places no requirements on the TOE to provide an identification and authentication capability to restrict these management functions to an administrative role, which implies that there are a number of ways a TOE vendor could be compliant. For example,

- The TOE contains no notion of an authorized administrator; anyone that can invoke the management utility can configure the TOE. In order to be compliant to the PP in this case, the

TOE vendor must provide instructions as part of the AGD_OPE/PRE guidance that detail the procedures an administrator would use to configure the Operational Environment such that only a subset of the authorized users of the TOE would be able to execute the management utility. For example, the guidance would describe configuring the access control mechanisms in the Operational Environment so that only the administrator-allowed users would be able to execute the management utility. This case reflects the baseline requirements of this PP.

- The TOE contains a notion of an authorized administrator (or set of administrators), but relies on the Operational Environment to perform the identification and authentication functions and then pass some indication to the TOE that can be matched to the internal TOE representation of an authorized administrator. In this case, the ST author will need augment the requirements (using the templates provided in Appendix C) to specify the capabilities provided by the TOE. The vendor will need to describe any configuration or settings in the Operational Environment needed to support the passing of the information to the TOE.
- The TOE contains its own identification and authentication capability that is used to determine which users of the system housing the hard disk are authorized to use the management functions provided by the TOE. In this case, the ST author will need to use the I&A template information provided in Appendix C in the body of the ST to specify this functionality.

### 1.1.4 Protocol Compliance

10 The TOEs meeting this PP will implement the Internet Engineering Task Force (IETF) Internet Protocol Security (IPsec) Security Architecture for the Internet Protocol, RFC 4301, as well as the IPsec Encapsulating Security Payload (ESP) protocol. IPsec ESP is specified in RFC 2406 and RFC 4303. The IPsec VPN Client will support ESP in either tunnel mode, transport mode, or both modes.

11 The IPsec VPN Client will use either the Internet Key Exchange (IKE)v1 protocol as defined in RFCs 2407, 2408, 2409, 4109 or the IKEv2 protocol as specified in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), and 4307 to authenticate and establish session keys with the VPN entities.

12 In order to show that the TSF implements the RFCs correctly, the evaluator shall perform the assurance activities documented in this PP. In future versions of this PP, assurance activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than is currently described in this publication.

# 2 Security Problem Definition

13 This PP is written to address the situation in which a remote user uses a public network to access a private network (e.g., the user's office network). Protection of network packets is desired as they cross the boundary between a public network and a private network. To protect the data in-transit from disclosure and modification, a VPN is created to establish secure communications. The VPN Client provides one end of the secure VPN tunnel and performs encryption and decryption of network packets in accordance with a VPN security policy negotiated between the VPN Client and a VPN Gateway.

14 The proper installation and configuration of the VPN Client are critical to its correct operation such that proper handling of the TOE by an administrator is also addressed.

15 This chapter identifies the following:

- IT related threats to the organization countered by the VPN Client;
- Environmental threats requiring controls to provide sufficient protection;
- Organizational security policies for the VPN Client as appropriate; and
- Significant assumptions about the VPN Client's operational environment.

## 2.1 Threats

16 This PP does not include requirements that can protect against an insider threat. Authorized users are not considered hostile or malicious and are trusted to follow appropriate guidance. Only authorized personnel should have access to the client device. Therefore, the primary threat agents are the unauthorized entities that try to gain access to the protected network. An entity is authorized if they can authenticate themselves to the private network thus establishing themselves as legitimate users of the network. In this situation, the TOE is the requesting entity that must authenticate. The established connection is subject to network attacks and must be protected from disclosure and modification. Likewise, the TOE must also ensure that it establishes a communication tunnel with a legitimate VPN Gateway and that the VPN Gateway is not masquerading as a trusted entity. Mutual authentication will prohibit connections with unauthorized entities. The TOE will protect itself from compromise caused by errors or malicious actions.

17 Improper negotiation of security policies or enforcing weak protocol options to establish a VPN connection is also a concern that could result in the disclosure or modification of user and TSF data. Protocol interoperability and mutual agreed upon security policies requiring strong encryption are imperative for establishing VPN protection.

18 Other threat agents include security related information that is not cleared when resources are reallocated; when sensitive values are no longer needed, access to these data must be prevented. The TOE must ensure that residual data are appropriately handled such that security related information is not accessible by other users/processes after it is used. Compromise of TSF data includes authentication data, session keys, security mechanisms, and the data the TOE protects. TOE or TSF data must be protected from inappropriate access and updates.

19 Network attacks, such as that described above against the TOE, are not the sole avenue for gaining unauthorized access and compromising security. Updating products is a common and necessary capability to ensure that changes to the threat environment are addressed; a common attack vector used involves attacking un-patched versions of software containing flaws. Timely application of patches

increases the likelihood that the product will be able to maintain and enforce its security policy. However, the updates must be from a trusted source; otherwise, an attacker can write their own "update" that contains malicious code of their choosing, such as a rootkit, bot, or other malware.

20 Once an adversary has access, regardless of the mechanism used to obtain it (network attacks, malicious code, taking advantage of errors in configuration, session hijacking, etc.), the TOE and its data have been compromised. Modification of audit record generation to hide any further nefarious actions taken on the TOE could mask potential problems as well as make it difficult to identify who caused the malicious action. Undetected actions may adversely affect the security of the TOE and may make it difficult to mitigate the problems caused. Note that audit review and storage are handled by the IT environment and are therefore outside the scope of this PP. However, it is assumed that this is done properly and securely to protect the TOE.

21 The following table lists the threats addressed by the VPN Client and the operational environment. The assumed level of expertise of the attacker for all the threats identified below is unsophisticated.

**Table 1: Threats**

| Threat | Description of Threat |
|---|---|
| T.TSF_FAILURE | Security mechanisms of the TOE may fail, leading to a compromise of the TSF. |
| T.UNAUTHORIZED_ACCESS | A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. |
| T.UNAUTHORIZED_UPDATE | A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE. |
| T.USER_DATA_REUSE | User data may be inadvertently sent to a destination not intended by the original sender. |

## 2.2 Organizational Security Policies

22 The Organizational Security Policies were selected because of their applicability to protect network packets crossing the boundary between a private network and a public network. The policies relating to procedures are also stated as assumptions. Those policies that do not have a formal reference are expected to be created and formalized subject to the policy description.

**Table 2: Organizational Security Policies**

| Policy | Policy Description |
|---|---|
| P.COMPATIBILITY | The TOE must meet Request for Comments (RFC) requirements for implemented protocols to facilitate inter-operability-with other network equipment using the same protocols. |

| Policy | Policy Description |
|---|---|
| P.CONFIGURABILITY | The TOE must provide the capability to configure security-relevant aspects of its operation. |

## 2.3 Assumptions

23 This section of the security problem definition shows the assumptions that are made on the operational environment in order to be able to provide security functionality. If the TOE is placed in an operational environment that does not meet these assumptions, the TOE may no longer be able to provide all of its security functionality. Assumptions can be on the physical environment, personnel, and connectivity of the operational environment.

**Table 3: TOE Assumptions**

| Assumption | Description of Assumption |
|---|---|
| A.NO_TOE_BYPASS | Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE. |
| A.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment. |
| A.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |

# 3 Security Objectives

24 The Security Objectives are the requirements for the Target of Evaluation (TOE) and for the Operational Environment derived from the threats, organizational security policies, and the assumptions in Section 2. Section 3 restates the Security Objectives for the TOE more formally as SFRs. The TOE is evaluated against the SFRs.

## 3.1 Security Objectives for the TOE

25 Table 4 identifies the Security Objectives for the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. The TOE meets these objectives by satisfying the security functional requirements.

**Table 4: Security Objectives for the TOE**

| Objective | Objective Description |
|---|---|
| O.AUTH_COMM | The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. |
| O.CRYPTOGRAPHIC_FUNCTIONS | The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of data that are transmitted outside the TOE and its host environment. |
| O.GW_AUTHENTICATION | The TOE will authenticate the VPN Gateway that it attempts to establish a security association with. |
| O.PROTOCOLS | The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or industry specifications to ensure interoperability. |
| O.RESIDUAL_INFORMATION_CLEARING | The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. |
| O.TOE_ADMINISTRATION | The TOE will provide mechanisms to allow administrators to be able to configure the TOE. |
| O.TSF_SELF_TEST | The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. |
| O.VERIFIABLE_UPDATES | The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source. |

## 3.2 Security Objectives for the Operational Environment

26 The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the

TOE). This part-wise solution is called the security objectives for the operational environment and consists of a set of statements describing the goals that the operational environment should achieve.

27 This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 2.3 are incorporated as Security Objectives for the Operational Environment. They levy additional requirements on the environment, which are largely satisfied through procedural or administrative measures. Table 5 identifies the security objectives for the environment.

**Table 5: Security Objectives for the Operational Environment**

| Objective | Objective Description |
|---|---|
| OE.NO_TOE_BYPASS | Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE. |
| OE.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the operational environment. |
| OE.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |

## 3.3 Security Objective Rationale

28 This section describes the rationale for the Security Objectives as defined in Section 3. Table 6 illustrates the mapping from Security Objectives to Threats and Policies.

**Table 6: Security Objectives to Threats and Policies Mappings**

| Threat/Policy | Objectives Addressing the Threat and Policies | Rationale |
|---|---|---|
| T.TSF_FAILURE<br>Security mechanisms of the TOE may fail, leading to a compromise of the TSF. | O.TSF_SELF_TEST<br>The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. | O. TSF_SELF_TEST counters this threat by ensuring that the TSF runs a suite of self tests to successfully demonstrate the correct operation of the TSF. |
| T.UNAUTHORIZED_ACCESS<br>A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. | O.AUTH_COMM<br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity.<br>O.CRYPTOGRAPHIC_FUNCTIONS<br>The TOE shall provide cryptographic functions (i.e., encryption/ decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.<br>O.GW_AUTHENTICATION<br>The TOE will authenticate the VPN Gateway that it attempts to establish a security association with.<br>O.TOE_ADMINISTRATION<br>The TOE will provide mechanisms to allow administrators to be able to configure the TOE. | O.AUTH_COMM and O.GW_AUTHENTICATION work to mitigate this threat by ensuring that the TOE identifies and authenticates VPN Gateways prior to communicating with that entity. The TOE must also be capable of sending its own credentials to the VPN Gateway to ensure mutual authentication prior to communication.<br>O.CRYPTOGRAPHIC_FUNCTIONS contributes to mitigating this threat by providing the underlying cryptographic functionality required by other protection mechanisms.<br>O.TOE_ADMINISTRATION requires the TOE to provide mechanisms to allow the TOE to be configured in a secure manner. |
| T.UNAUTHORIZED_UPDATE<br>A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE. | O.VERIFIABLE_UPDATES<br>The TOE will provide the capability to ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source. | O.VERIFIABLE_UPDATES ensures that the administrator can confirm the update. |
| T.USER_DATA_REUSE | O.RESIDUAL_INFORMATION_CLEARING | O.RESIDUAL_INFORMATION_CLEARING counters this threat by |

| Threat/Policy | Objectives Addressing the Threat and Policies | Rationale |
|---|---|---|
| User data may be inadvertently sent to a destination not intended by the original sender. | The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. | ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. |
| P.COMPATIBILITY<br>The TOE must meet Request for Comments (RFC) requirements for implemented protocols to facilitate inter-operation with other network equipment using the same protocols. | O.PROTOCOLS<br>The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability, that also support communication with a centralized audit server and a RADIUS authentication server. | O.PROTOCOLS satisfies this policy by requiring that standardized protocols are implemented in the TOE to ensure interoperability among IT entities using the same protocols. |
| P.CONFIGURABILITY<br>The TOE must provide the capability to configure security-relevant aspects of its operation. | O.TOE_ADMINISTRATION<br>The TOE will provide mechanisms to allow administrators to be able to configure the TOE. | O.TOE_ADMINISTRATION satisfies the policy by ensuring that the TOE provides the mechanisms needed to security configure the TOE. |

29 Table 7 illustrates the mapping from Security Objectives to Assumptions.

**Table 7: Security Objectives to Assumptions Mappings**

| Assumption | Objectives Addressing the Assumption | Rationale |
|---|---|---|
| A.NO_TOE_BYPASS<br>Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE. | OE.NO_TOE_BYPASS<br>Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE. | OE.NO_TOE_BYPASS ensures that all information that flows onto the network passes through the TOE. |
| A.PHYSICAL<br>Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment. | OE.PHYSICAL<br>Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the operational environment. | OE.PHYSICAL ensures the TOE, the TSF data, and protected user data is protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment. |
| A.TRUSTED_ADMIN<br>TOE Administrators are trusted to | OE.TRUSTED_ADMIN<br>TOE Administrators are trusted to | OE.TRUSTED_ADMIN ensures the administrators are properly trained and the administrative guidance |

| Assumption | Objectives Addressing the Assumption | Rationale |
|---|---|---|
| follow and apply all administrator guidance in a trusted manner. | follow and apply all administrator guidance in a trusted manner. | instructs the administrator how to properly configure the environment and TOE to avoid mistakes. |

# 4 Security Requirements and Rationale

30 The Security Requirements are divided into functional requirements and assurance requirements. The SFRs are a formal instantiation of the Security Objectives and are provided with application notes in Section 4.1. They are usually at a more detailed level of abstraction, but they have to be a complete translation (the security objectives must be completely addressed). The CC requires this translation into a standardized language for several reasons:

- To provide an exact description of what is to be evaluated. As security objectives for the TOE are usually formulated in natural language, translation into a standardized language enforces a more exact description of the functionality of the TOE.
- To allow comparison between two STs. As different ST authors may use different terminology in describing their security objectives, the standardized language enforces using the same terminology and concepts. This allows easy comparison.

31 The Security Assurance Requirements (SARs) are typically boilerplate that is inserted and listed separately from the SFRs; the Common Evaluation Methodology (CEM) is then consulted during the evaluation based on the SARs chosen. A more tailored approach is taken in this PP based on the new model for Standard Protection Profiles. While the SARs are still listed for context and completeness in Section 4.3, the activities that an evaluator needs to perform for this TOE with respect to each SFR and SAR are detailed in "Assurance Activities" paragraphs. Assurance Activities are normative descriptions of activities that must take place in order for the evaluation to be complete. Assurance Activities are located in two places in this PP; those that are associated with specific SFRs are located in Section 4.1, while those that are independent of the SFRs are detailed in Section 4.3.

32 For the activities associated directly with SFRs, after each SFR one or more Assurance Activities is listed detailing the activities that need to be performed to achieve the assurance provided for this technology.

33 For the SARs that require activities that are independent of the SFRs, Section 4.3 indicates the additional Assurance Activities that need to be accomplished, along with pointers to the SFRs for which specific Assurance Activities associated with the SAR have been written.

34 Future iterations of the Protection Profile may provide more detailed Assurance Activities based on lessons learned from actual product evaluations.

## 4.1 Security Functional Requirements

35 This section identifies the SFRs for the TOE that are specific to the security functionality provided by the TOE and distinguishes the VPN Client from other TOEs. The focus areas of the SFRs are related to audit, cryptography, security management, self-tests, and communication with authorized external IT entities (e.g., VPN Gateway).

**Table 8: TOE Security Functional Requirements**

| Functional Class | Functional Components |
|---|---|
| Cryptographic support Class (FCS) | FCS_CKM.1 (1)Cryptographic key generation (Asymmetric Keys) |
| | FCS_CKM.1(2) Cryptographic Key Generation (for asymmetric keys - IKE) |
| | FCS_CKM_EXT.4 Cryptographic Key Zeroization |
| | FCS_COP.1(1) Cryptographic operation (Data Encryption/Decryption) |
| | FCS_COP.1(2) Cryptographic operation (Cryptographic Signature) |
| | FCS_COP.1(3) Cryptographic operation (Cryptographic Hashing) |
| | FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication) |
| | FCS_IPSEC_EXT Extended: Internet Protocol Security (IPsec) Communications |
| | FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation) |
| User Data Protection Class (FDP) | FDP_RIP.2 Full residual information protection |
| Identification and Authentication Class (FIA) | FIA_X509_EXT.1 Extended: X.509 Certificates |
| Security Management Class (FMT) | FMT_SMF.1 Specification of Management Functions |
| Protection of the TSF (FPT) | FPT_TST_EXT.1 Extended: TSF Testing |
| | FPT_TUD_EXT.1 Extended: Trusted Update |
| Trusted Path/Channels (FTP) | FTP_ITC.1 Inter-TSF trusted channel |

### 4.1.1 Class: Cryptographic Support (FCS)

**FCS_CKM.1 Cryptographic Key Generation (Asymmetric Keys)**

FCS_CKM.1.1 **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

- *NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;*
- *NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard")*
- *[selection: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes, no other*]

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits*.

*Application Note:*

36 *This component requires that the TOE be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols used by the TOE.*

37 *Since the domain parameters to be used are specified by the requirements of the protocol in this PP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies with the protocols specified in this PP.*

***Assurance Activity:***

38 *The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

39 *In order to show that the TSF complies with 800-56A and/or 800-56B, depending on the selections made, the evaluator shall ensure that the TSS contains the following information:*

- *The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.*
- *For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;*
- *For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to "shall" or "should" statements shall be described;*

*Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.*

**FCS_CKM.1 (2) Cryptographic Key Generation (for asymmetric keys - IKE)**

FCS_CKM.1.1(2) **Refinement**: The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a:

[selection, choose at least one of:

- FIPS PUB 186-3, “Digital Signature Standard (DSS)”, Appendix B.3 for RSA schemes;

- FIPS PUB 186-3, “Digital Signature Standard (DSS)”, Appendix B.4 for ECDSA schemes and implementing “NIST curves” P-256, P-384 and [selection: P-521, no other curves];

- ANSI X9.31-1998, Appendix A.2.4 Using AES for RSA schemes]

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits*.

*Application Note: The ANSI X9.31-1998 option will be removed from the selection in a future publication of this document. Presently, the selection is not exclusively limited to the FIPS PUB 186-3 options in order to allow industry some further time to complete the transition to the modern FIPS PUB 186-3 standard.*

*The keys that are required to be generated by the TOE through this requirement are intended to be used for the authentication of the VPN entities during the IKE (either v1 or v2) key exchange. While it is required that the public key be associated with an identity in an X509v3 certificate, this association is not required to be performed by the TOE, and instead is expected to be performed by a Certificate Authority in the Operational Environment.*

*As indicated in FCS_IPSEC_EXT.1, the TOE is required to implement support for RSA or ECDSA (or both) for authentication.*

*The generated key strength of 2048-bit RSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, “Recommendation for Key Management” for information about equivalent key strengths.*

***Assurance Activity:***

*The evaluator shall use the key pair generation portions of "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)" and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

*The evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with FIPS PUB 186-3, the evaluator shall ensure that the TSS contains the following information:*

- *The TSS shall list all sections of Appendix B to which the TOE complies.*
- *For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;*

- *For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described;*

*Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.*

**FCS_CKM_EXT.4 Cryptographic Key Zeroization**

FCS_CKM_EXT.4.1 **Refinement:** The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

*Application Note:*

40 *Any security related information (such as keys, authentication data, and passwords) must be zeroized when no longer in use to prevent the disclosure or modification of security critical data.*

41 *The zeroization indicated above applies to each intermediate storage area for plaintext key/CSP (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/CSP to another location.*

42 *Since the TOE does not necessarily include the host IT environment, the extent of this capability is necessarily somewhat limited. For the purposes of this requirement, it is sufficient for the TOE to invoke the correct underlying functions of the host to perform the zeroization--it does not imply that the TOE has to include a kernel-mode memory driver to ensure the data are zeroized.*

***Assurance Activity:***

43 *The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write"). If a read-back is done to verify the zeroization, this shall be described as well.*

44

## Cryptographic Operation (FCS_COP)

**FCS_COP.1(1) Cryptographic Operation (Data Encryption/Decryption)**

FCS_COP.1.1(1) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm *AES operating in* **GCM, CBC,** [**assignment:** ***one or more modes, no other modes***]] and cryptographic key sizes 128-bits, 256-bits, and [**selection: 192 bits, no other key sizes**] that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**

- **NIST SP 800-38D, NIST SP 800-38A [selection:, NIST SP 800-38B, NIST SP 800-38C, NIST SP 800-38E, no other standards]**

*Application Note:*

45 *This PP requires the modes GCM and CBC to be used in the IPsec and IKE protocols (FCS_IPSEC_EXT.1.4, FCS_IPSEC_EXT.1.6). Therefore, the FCS_COP.1.1(1) element in the NDPP has been specified here to ensure the ST Author includes these two modes to be consistent with the IPsec requirements.*

***Assurance Activity:***

46 *The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

**FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)**

**FCS_COP.1.1(2)** **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a:

- **[selection, choose at least one of: *RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets* FIPS PUB 186-2 or FIPS PUB 186-3, “Digital Signature Standard”,**
- ***Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets* FIPS PUB 186-3, “Digital Signature Standard” with “NIST curves” P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-3, “Digital Signature Standard”)].**

***Assurance Activity:***

47 *The evaluator shall use the signature generation and signature verification portions of "The Elliptic Curve Digital Signature Algorithm Validation System” (ECDSA2VS), and "The RSA Validation System” (RSA2VS) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e. FIPS PUB 186-3). This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

**FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing)**

FCS_COP.1.1(3) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [**selection: SHA-1, SHA-256, SHA-384] and message digest sizes** [**selection: 160, 256, 384**] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

*Application Note:*

48 *The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.*

***Assurance Activity:***

49 *The evaluator shall use "The Secure Hash Algorithm Validation System (SHAVS)" as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

**FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication)**

FCS_COP.1.1(4) **Refinement:** The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm **HMAC-** [selection: SHA-1, SHA-256, SHA-384],-key size **[assignment: key size (in bits) used in HMAC], and message digest size of** [selection: 160, 256, 384] **bits** that meet the following: **FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-3, "Secure Hash Standard"**.

*Application Note:*

50 *The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if HMAC-SHA-256 is chosen, then the only valid message digest size selection would be 256 bits.*

51 *The message digest size above corresponds to the underlying hash algorithm used. Note that truncating the output of the HMAC following the hash calculation is an appropriate step in a variety of applications. This does not invalidate compliance with this requirement, however, the ST should state that truncation is performed, the size of the final output, and the standard to which this truncation complies.*

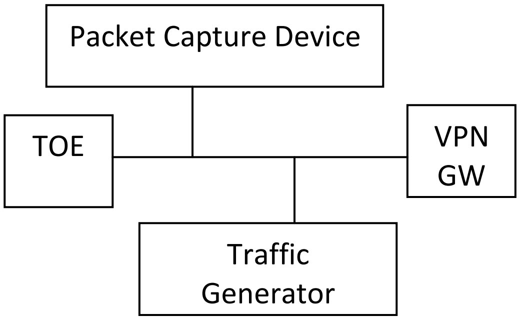
***Assurance Activity:***

52 *The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

### Extended: Internet Protocol Security (FCS_IPSEC_EXT)

53 In order to show that the TSF implements the RFCs correctly, the evaluator shall perform the assurance activities listed below. In future versions of this PP, assurance activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than is currently described in this publication.

54 The TOE is required to use the IPsec protocol to establish connections used to communicate with a VPN Gateway.



*The evaluators shall minimally create a test environment equivalent to the test environment illustrated above. It is expected that the traffic generator is 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 evaluators must provide Justification for any differences in the test environment.*

**FCS_IPSEC_EXT.1**

**Extended: Internet Protocol Security (IPsec) Communications**

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

***Assurance Activity:***

***TSS***

*Nothing is done in addition to determining that the TOE's implementation is conformant to RFC 4301 as described above.*

***Guidance***

*The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for DISCARD, BYPASS and PROTECT.*

***Test***

*The evaluator uses the operational guidance to configure the TOE to carry out the following tests:*

*Test 1: The evaluator shall configure the TOE's SPD such that there is a rule for DISCARD, BYPASS, PROTECT. The selectors used in the construction of the rule shall be different such that the evaluator can* *send in three network packets with the appropriate fields in the packet header that each packet will match one of the three rules. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packet was dropped, allowed through without modification, was encrypted by the IPsec implementation.*

*Test 2: The evaluator shall devise two equal SPD entries with alternate operations – BYPASS and PROTECT. The entries should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first entry is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.*

*Test 3: The evaluator shall repeat the procedure above, except that the two entries should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.*

FCS_IPSEC_EXT.1.2 The TSF shall implement [selection, choose at least one of: tunnel mode, transport mode].

***Assurance Activity:***
***TSS***
*The evaluator checks the TSS to ensure it states that the TOE can operate in tunnel mode and/or transport mode (as selected).*
***Guidance***
*The evaluator shall confirm that the operational guidance instructs the Administrator how the TOE is configured in each mode selected.*
***Test***
*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 GW to operate in tunnel mode. The evaluator configures the TOE and the VPN GW 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 GW peer. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the 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 a VPN GW to operate in transport mode. The evaluator configures the TOE and the VPN GW 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 VPN GW. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the 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.

***Assurance Activity:***
***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 checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.*

***Test***

*Test 1: The evaluator shall configure the TOE's SPD, such that it has entries that contain operations that DISCARD, BYPASS, and PROTECT network packets. The evaluator also configures the TOE so that all auditable events with respect to FCS_IPSEC_EXT.1 are enabled. 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 to the TOE. The evaluator should observe that the network packet is passed by the TOE to the proper destined 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 to the TOE, and observes that the packet was not permitted to flow to any of the TOE's interfaces. The evaluator shall verify that an audit record is generated that specifies that the packet was discarded as expected.*

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, AES-GCM-256 as specified in RFC 4106, [selection: AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, no other algorithms].

***Assurance Activity:***

***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 FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication).*

***Guidance***

*The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the AES-GCM-128, and AES-GCM-256 algorithms, and if either AES-CBC-128 or AES-CBC-256 have been selected the guidance instructs how to use these as well.*

***Test***

*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 The TSF shall implement the protocol: [selection, choose at least one of: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].

***Assurance Activity:***

***TSS***

*The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.*

***Guidance***

*The evaluator checks the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test.*

***Test***

*Test 1: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.*

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection, choose at least one of: IKEv1, IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

***Assurance Activity:***

***TSS***

*The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/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 ensures that the operational guidance describes how the TOE can be configured to use the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test.*

***Test***

*Test 1: The evaluator shall configure the TOE to use AES-CBC-128 to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using AES-CBC-128. The evaluator will consult the audit trail to confirm the algorithm was that used in the negotiation.*

FCS_IPSEC_EXT.1.7 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

***Assurance Activity:***

***TSS***

*The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.*

***Guidance***

*If the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.*

***Test***

*Test 1 (conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.*

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection: IKEv2 SA lifetimes can be configured by [selection: an Administrator, VPN Gateway] based on number of packets/number of bytes or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs, IKEv1 SA lifetimes can be configured by an [selection: an Administrator, VPN Gateway] based on number of packets/number of bytes or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours 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 client or a VPN gateway that pushes the SA lifetime down to the client 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.*

***Assurance Activity:***

***TSS***

*How the lifetimes are established and enforced is described in the RFCs and the evaluator examines the TSS as stated at the beginning of this section.*

***Guidance***

*The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. The evaluator ensures that either the Administrator or VPN Gateway are able to configurable Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured.*

***Test***

*When testing this, the evaluator needs to ensure that both sides 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:*

*Test 1: The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # 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 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.*

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.

***Assurance Activity:***

*The evaluator shall check to ensure that, for each DH group supported by the TSF, 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 PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.*

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*] .

***Assurance Activity:***

*The evaluator shall check to ensure that, for each DH group supported by the TSF, 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 PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.*

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [selection: 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: other DH groups that are implemented by the TOE], no other DH groups].

***Assurance Activity:***

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

FCS_IPSEC_EXT.1.12 The TSF shall ensure that all IKE protocols perform peer authentication using a [selection, choose at least one of: *RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

***Assurance Activity:***

*If the TOE performs the capabilities specified in Appendix C for X.509 certificates (i.e., protected certificate store, or certificate path validation) the requirements associated with those capabilities will be placed in the body of the PP in the FIA_X509_EXT.1 component, and the associated assurance activities will be placed here by the ST Author.*

***TSS***

*The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1(2) Cryptographic Operations (for cryptographic signature).*

*If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.*

***Guidance***

*The evaluator ensures the operational guidance describes how to set up the TOE to use the cryptographic algorithms RSA and/or ECDSA.*

*In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operation 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 and marked "trusted".*

***Test***

*For efficiency sake, the testing that is performed here has been combined with aspects of the testing for FIA_X509_EXT.1 Extended: X.509 Certificates, specifically FIA_X509_EXT.1.4, and FIA_X509_EXT.1.5. The following tests shall be repeated for each peer authentication protocol selected in the FCS_IPSEC_EXT.1.12 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.*

*Test 2: The evaluator shall use a certificate signed using the RSA or ECDSA algorithm to authenticate the remote peer during the IKE exchange. This test ensures the remote peer has the certificate for the trusted CA that signed the TOE's certificate and it will do a bit-wise comparison on the DN. This bit-wise comparison of the DN ensures that not only does the peer have a certificate signed by the trusted CA, but the certificate is from the DN that is expected. The evaluator will configure the TOE to associate a certificate (e.g., a certificate map in some implementations) with a VPN connection. This is what the DN is checked against.*

*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 draft of the EP, 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: The evaluator shall test that given a signed certificate from a trusted CA, that when the DN does not match – any of the four fields can be modified such that they do not match the expected value, that an SA does not get established.*

*Test 5: The evaluator shall ensure that the TOE is configurable to either establish an SA, or not establish an SA if a connection to the certificate validation entity cannot be reached. For each method selected for*

*certificate validation, the evaluator attempts to validate the certificate – for the purposes of this test, it does not matter if the certificate is revoked or not. For the "mode" where an SA is allowed to be established, the connection is made. Where the SA is not to be established, the connection is refused.*
*Test 6 [conditional]: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection between the TOE and the VPN GW peer. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.*

FCS_IPSEC_EXT.1.13 The TSF 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.

***Assurance Activity:***

***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/or 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 this is protecting the negotiation.*
***Guidance***
*The evaluator simply follows the guidance to configure the TOE to perform the following tests.*
***Test***
*Test 1: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.*
*Test 2: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more 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: This test shall be performed for each version of IKE supported by the TOE. 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: This test shall be performed for each version of IKE supported by the TOE. 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.*

*Application Note:*

55 *FCS_IPSEC_EXT.1.7* is only applicable if IKEv1 is selected.

56 *FCS_IPSEC_EXT.1.8: 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 hardcoded limits, but in this case it is required than 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 administrative 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). It is also appropriate to refine the requirement such that SA lifetime management and enforcement occurs external to the TOE (i.e.: on a VPN Gateway), however, even when the requirement is refined in this manner, the evaluator shall conduct the associated assurance activities described above.*

57 *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).*

58 *FCS_IPSEC_EXT.1.11: The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. In future versions of this PP, DH Group 20 (384-bit RandomECP) will be required. 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.*

59 *FCS_IPSEC_EXT.1.12: At least one public-key-based Peer Authentication method is required for conformant TOEs; one or more of the public key schemes is chosen by the ST author to reflect what is implemented by the TOE. 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, 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).*

60 *FCS_IPSEC_EXT.1.13: The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for a TOE to allow this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the OPE documentation) must enable this functionality.*

## Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT)

### FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C; X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulates entropy from [selection: choose one of:

one or more independent hardware-based noise sources, one or more independent software-based noise sources, a combination of hardware-based and software-based noise sources].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

*Application Note:*

61 *NIST Special Pub 800-90, Appendix C describes the minimum entropy measurement that will probably be required in future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of this PP.*

62 *For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90 or 140-2 Annex C). For the second selection, the ST author indicates how the client collects entropy for the RBG.*

63 *SP 800-90 contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90 is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CT_DRBG are allowed. While any of the curves defined in 800-90 are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.*

64 *Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.*

65 *The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.*

66 *In the future, most of the requirements described in A Method for Entropy Source Testing: Requirements and Test Suite Description will be required by this PP. The follow Assurance Activities currently reflect only that subset of activities that are required.*

***Assurance Activity:***

67 *The evaluator shall review the TSS section to determine the version number of the product containing the RBG(s) used in the TOE. The evaluator shall also confirm that the TSS describes the noise source or sources from which entropy is gathered. The evaluator will further verify that all of the underlying functions and parameters used in the RBG are listed in the TSS.*

68 *The evaluator shall verify that the TSS contains a description of the RBG model, including the method for obtaining entropy input, as well as identifying the entropy source(s) used, how entropy is produced/gathered from each source, and how much entropy is produced by each entropy source. The evaluator shall also ensure that the TSS describes the entropy source health tests, a rationale for why the health tests are sufficient to determine the health of the entropy sources, and known modes of entropy*

*source failure. Finally, the evaluator shall ensure that the TSS contains a description of the RBG outputs in terms of the independence of the output and variance with time and/or environmental conditions.*

69 *Regardless of the standard to which the RBG is claiming conformance, the evaluator performs the following test:*

- *Test 1: The evaluator shall determine an entropy estimate for each entropy source by using the Entropy Source Test Suite. The evaluator shall ensure that the TSS includes an entropy estimate that is the minimum of all results obtained from all entropy sources.*

70 *The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.*

**Implementations Conforming to FIPS 140-2, Annex C**

71 *The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluators shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.*

72 *The evaluators shall perform a Variable Seed Test. The evaluators shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluators ensure that the values returned by the TSF match the expected values.*

73 *The evaluators shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluators then invoke the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluators ensure that the 10,000th value produced matches the expected value.*

**Implementations Conforming to NIST Special Publication 800-90**

74 *The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.*

75 *If the RNG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "Generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).*

76 *If the RNG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The*

*evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.*

77 *The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.*

> ***Entropy input:** the length of the entropy input value must equal the seed length.*
> ***Nonce:** If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.*
> ***Personalization string:** The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.*
> ***Additional input:** the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.*

### 4.1.2 Class: User Data Protection (FDP)

#### Residual Information Protection (FDP_RIP)

| | |
|---|---|
| **FDP_RIP.2** | **Full Residual Information Protection** |
| FDP_RIP.2.1 | The TSF shall enforce that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects. |

*Application Note:*

78 *This requirement ensures, for example, that protocol data units (PDUs) are not padded with residual information such as cryptographic key material. The ST author uses the selection to specify when previous information is made unavailable.*

***Assurance Activity:***

79 *“Resources” in the context of this requirement are network packets being sent through (as opposed to “to”, as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.*

### 4.1.3 Class: Identification and Authentication (FIA)

80 The baseline requirements for the TOE are fairly limited with respect to I&A, since no formal administrative or general purpose users are defined. The extent of the I&A required to be performed by the TOE relates to the authentication done at the machine level when establishing the IPsec connection. These I&A requirements are specified in the FCS_IPSEC_EXT.1 component to keep requirements on the IPsec protocol grouped together for understandability as well as for ease of authoring and applying assurance activities. Therefore, the requirements in this section cover only the credentials that are used by the protocols specified in this PP.

81 It is important to note that there are two elements of the **FIA_X509_EXT.1** component that exist in Appendix C.1. Those two elements deal with the protection of the certificates, where it is envisioned the underlying operating system will by employed to provide some level of protection, and the validity checking of a given certificate. It is possible that the VPN Client performs the validity check, in which case the requirement (FIA_X509_EXT.1.5) would be moved to the body of the PP in this section. It may be the case that the TOE relies on the operating system to perform this check, in which case the requirement remains in C.1, with a clear indication that the environment performs the check and provides a result to the TOE.

## X509 Certificates (FIA_X509_EXT)

### FIA_X509_EXT.1 Extended: X.509 Certificates

FIA_X509_EXT.1.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec connections.

*Application Note:*

82 *It should be noted that RFC 5280 defines certificate validation and certification path validation requirements that must be implemented by the TOE as per this requirement.*

FIA_X509_EXT.1.3 The TSF shall provide the capability for Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

FIA_X509_EXT.1.4 The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

*Application Note:*
*The public key referenced in FIA_X509_EXT.1.4 is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1(2).*

FIA_X509_EXT.1.8 The TSF shall not establish an SA if a certificate or certificate path is deemed invalid.

FIA_X509_EXT.1.9 The TSF shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

FIA_X509_EXT.1.10 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall, at the option of the administrator, establish an SA or disallow the establishment of an SA..

*Application Note:*

*The intent of FIA_X509_EXT.1.8 is that the TOE is configurable to allow or disallow session establishment if the TOE cannot connect to an entity responsible for providing certificate validation information. For instance, if a CRL cannot be obtained because a machine is down, or the network path is broken, the administrator may elect to configure the TOE to allow sessions to continued to be established, rather than terminate the TOE's ability to establish any new SAs because it cannot reach the CA.*

***Assurance Activity:***

83 *The evaluator shall check the administrative guidance to ensure that it describes how to configure the TOE to use X.509 certificates. This description includes how to load certificates into the TOE, how to generate keys and then make a CRM request to get a certificate for the TOE itself. The description also instructs the administrator how to configure the TOE to either allow or disallow an SA to be established if a decision cannot be made regarding the validity of a certificate being used to authenticate the gateway. In some configurations it may be necessary for the TOE to have established a connection via the VPN Gateway in order to obtain certificate validity information.*

*The evaluator shall examine the TSS to determine that it describes how the TOE implements certificates to satisfy the requirements. This description includes what aspects are performed by the TOE, and which are allocated to the operational environment.*

*The testing to ensure the requirements are satisfied is performed in conjunction with the IPsec requirement FCS_*IPSEC_EXT.1.12.

### 4.1.4 Class: Security Management (FMT)

84 As indicated in Section 1 of this PP, the TOE is not required to maintain a separate management role. They are, however, required to provide functionality to configure certain aspects of TOE operation that should not be available to the general user population. If the TOE does provide some degree of administrative control, then the appropriate requirements from Appendix C should be used in the ST.

#### Specification of Management Functions (FMT_SMF)

**FMT_SMF.1** **Specification of Management Functions**

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- **Specify the security associations that shall be proposed and accepted during the IKE negotiations,**
- **Configuration of IKE protocol version(s) used,**
- **Configure IKE authentication techniques used,**

- **Configure the cryptoperiod for the established session keys. The unit of measure for configuring the cryptoperiod shall be no greater than an hour,**
- **Configure certificate revocation check,**
- **Specify the algorithm suites that may be proposed and accepted during the IPsec exchanges,**
- **Specify authentication methods for peer-to-peer connections, if allowed,**
- **ability to update the TOE, and to verify the updates,**
- **ability to configure all security management functions identified in other sections of this PP,**
- **[assignment: any additional management functions].**

*Application Note:*

85 *For installation, the VPN Client relies on the IT environment to authenticate the administrator to the client machine.*

86 *For the function configure the cryptoperiod for the established session keys, the unit of measure for configuring the cryptoperiod shall be no greater than an hour. For example: units of measure in seconds, minutes and hours are acceptable and units of measure in days or greater are not acceptable.*

87 *There may be some instances where a VPN Gateway "pushes" configuration information down to the VPN Clients. This is an acceptable form of management, the ST Author simply must make clear in the ST what management functions are performed on the platform the VPN Client resides, and which are performed by the VPN Gateway. It may be the case that the functions overlap (i.e., can be done by an end-user on the platform or by the Gateway) and this is fine as long as the ST is clear and the guidance documentation describes how to perform the functions.*

88 ***Assurance Activity:***

89 *The evaluator shall check to make sure that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function. The evaluator shall test the TOE's ability to provide the management functions by configuring the TOE and testing each option listed in the requirement above.*

90 *As stated in the application note, a TOE may be configured either locally on the platform, or remotely by a VPN Gateway. The ST will clearly state which functions can be performed locally and remotely. The guidance documentation will describe how this is performed as well. The evaluator is expected to test this functions in all the ways in which the ST and guidance documentation state the configuration can be managed.*

91 *Note that the testing here may be accomplished in conjunction with the testing of other requirements, such as FCS_IPSEC_EXT.1.*

### 4.1.5 Class: Protection of the TSF (FPT)

#### Extended: TSF Self Test (FPT_TST_EXT)

**FPT_TST_EXT.1 Extended: TSF Self Test**

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2 The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic services.

*Application Note:*

*While the TOE is typically a software package running in the IT Environment, it is still capable of performing the self-test activities required above. It should be understood, however, that there is a significant dependency on the host environment in assessing the assurance provided by the tests mentioned above (meaning that if the host environment is compromised, the self tests will not be meaningful).*

***Assurance Activity:***

92 *The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.*

93 *The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases. The evaluator shall perform the following tests:*

- *Test 1: The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful.*
- *Test 2: The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.*

## Extended: Trusted Update (FPT_TUD_EXT.1)

**FPT_TUD_EXT.1 Extended: Trusted Update**

FPT_TUD_EXT.1.1 The TSF shall provide authorized administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide authorized administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: published hash, no other functions] prior to installing those updates.

*Application Note:*

94 *The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2). The published hash referenced is generated by one of the functions specified in FCS_COP.1(3).*

***Assurance Activity:***

95 *Updates to the TOE are signed by an authorized source and may also have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:*

- *Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.*

- *Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.*

## 4.1.6 Class: Trusted Path/Channels (FTP)

### Trusted Channel (FTP_ITC)

**FTP_ITC.1** **Inter-TSF trusted channel**

FTP_ITC.1.1 **Refinement:** The TSF shall **use IPsec** to provide a **trusted** communication channel between itself and **a VPN Gateway** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit *the TSF* to initiate communication via the trusted channel.

FTP_ ITC.1.3 The TSF shall initiate communication via the trusted channel *for all traffic traversing that connection*.

*Application Note:*

96 *The intent of the above requirement is to use the cryptographic protocols identified in the requirement to protect communications between the TOE and a VPN Gateway, both of which act as peers in the protocol sense.*

97 *The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.*

***Assurance Activity:***

98 *The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to an access point in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specification. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to the access point, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:*

- *Test 1: The evaluators shall ensure that the TOE is able to initiate communications with a VPN Gateway using the protocols specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- *Test 2: The evaluator shall ensure, for each communication channel with a VPN Gateway, the channel data is not sent in plaintext.*
- *Test 3: The evaluator shall ensure, for each communication channel with a VPN Gateway, modification of the channel data is detected by the TOE.*
- *Test 4: The evaluators shall physically interrupt the connection from the TOE to the a VPN Gateway. The evaluators shall ensure that subsequent communications are appropriately protected, at a minimum in the case of any attempts to automatically resume the connection or connect to a new access point.*

99 *Further assurance activities are associated with the specific protocols.*

## 4.2 Rationale for Security Functional Requirements

100 This section describes the rationale for the TOE Security Functional Requirements as defined in Section 4.1. Table 10 illustrates the mapping from Security Functional Requirements to Security Objectives with a corresponding rationale that the objective is addressed by the requirement.

101 The Security Target (ST) provided by the vendor also contains a security requirements rationale, consisting of two sections:

- a tracing that shows which SFRs address which security objectives for the TOE;
- a set of justifications that shows that all security objectives for the TOE are effectively addressed by the SFRs. (per CC part 1, Section B7).

Table 9: Rationale for TOE Security Functional Requirements

| Objective | Requirement Addressing the Objective | Rationale |
|---|---|---|
| O.AUTH_COMM<br>The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. | FCS_CKM.1<br>FCS_COP.1(1)<br>FCS_COP.1(2)<br>FCS_IPSEC_EXT.1<br>FIA_PSK_EXT.1<br>FIA_X509_EXT.1<br>FTP_ITC.1 | FTP_ITC.1 (and the supporting requirements FCS_CKM.1, FCS_COP.1(1), FCS_COP.1(2), FCS_IPSEC_EXT.1, FIA_PSK_EXT.1, and FIA_X509_EXT.1) require the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification. This is done cryptographically using the protocols specified by the requirements; these protocols provide the assured mutual identification of the endpoints and protection of the channel data. |
| O.CRYPTOGRAPHIC_FUNCTIONS<br>The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of data that are transmitted outside the TOE and its host environment. | FCS_CKM.1<br>FCS_CKM_EXT.4<br>FCS_COP.1(1)<br>FCS_COP.1(2)<br>FCS_COP.1(3)<br>FCS_COP.1(4)<br>FCS_RBG_EXT.1<br>FIA_X509_EXT.1 | FCS_CKM.1 generates asymmetric key. This key are used by for ephemeral key generation for IPSEC and potentially other public-key-based key agreement schemes.<br>FCS_CKM_EXT.4 provides the functionality for ensuring key and key material is zeroized. Since the TOE will in most cases be a software entity running on the host, the extent of this requirement is to make sure that the software invokes appropriate functions to clear the data; the host will ultimately be responsible for making sure the data are clear.<br>FCS_COP.1(1) specifies that AES be used to perform encryption and decryption operations for the various protocols specified in the PP. |

| | | |
|---|---|---|
| | | FCS_COP.1(2) requires a digital signature capability be implemented in the TOE for trusted updates and certificate operations associated with the protocols used to protect the traffic.<br>FCS_COP.1(3) and FCS_COP.1(4) require that the TSF provide hashing services using an implementation of the Secure Hash Algorithm algorithms for data integrity verification and non-data integrity operations.<br>FIA_X509_EXT.1 requires that the certificates used to support many of the cryptographic operations previously mentioned conform to an appropriate standard.<br>FCS_RBG_EXT.1 requires that a robust random bit generation capability be present. |
| O.PEER_AUTHENTICATION<br>The TOE will authenticate each peer TOE that attempts to establish a security association with the TOE. | FCS_IPSEC_EXT.1 | FCS_IPSEC_EXT.1 specifies that the TOE must implement IPsec using the Internet Key Exchange protocol. By implementing this protocol, the TOE will establish a secure, authenticated channel with each peer TOE for purposes of establishing a security association, which includes the establishment of a cryptographic key, algorithm and mode to be used for all communication. It is possible to establish multiple security associations between two peer TOEs, each with its own cryptographic key. Authentication may be via a digital signature and optionally a pre-shared key. |
| O.PROTOCOLS<br>The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability. | FCS_IPSEC_EXT.1<br>FTP_ITC.1 | FCS_IPSEC_EXT.1 and FTP_ITC.1 reference the standards (and indicate any restrictions on those standards) applicable to the protocol they require to be implemented. |
| O.RESIDUAL_INFORMATION_CLEARING<br>The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. | FCS_CKM_EXT.4<br>FDP_RIP.2 | FCS_CKM_EXT.4 ensures the destruction of any cryptographic keys when no longer needed.<br>FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly |

| | | |
|---|---|---|
| | | granted access to the data. For this TOE it is critical that the memory used to build network packets is either cleared or that some buffer management scheme be employed to prevent the contents of a packet being disclosed in a subsequent packet (e.g., if padding is used in the construction of a packet, it must not contain another user's data or TSF data). |
| O.SYSTEM_MONITORING<br>The TOE will provide the capability to generate audit data. | FAU_GEN.1<br>FAU_SEL.1 | FAU_GEN.1 defines the set of events that the TOE must be capable of recording, while FAU_SEL.1 allows the administrator to configure which auditable events will be recorded in the audit trail. |
| O.TOE_ADMINISTRATION<br>The TOE will provide mechanisms to allow administrators to be able to configure the TOE. | FAU_SEL.1<br>FMT_SMF.1 | FAU_SEL.1 requires the capability to configure the auditable events to be recorded, while FMT_SMF.1 provides configuration requirements for other parts of the TOE. As mentioned in the introduction, the TOE is not required to provide an administrative role, but the TOE in combination with the IT Environment must be capable of restricting these functions to a subset of the general users of the host machine. |
| O.TSF_SELF_TEST<br>The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. | FPT_TST_EXT.1 | FPT_TST_EXT.1 requires the TOE to provide a suite of self tests to assure the correct operation of the TSF, and to detect integrity problems in its stored executable. |
| O.VERIFIABLE_UPDATES<br>The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source. | FCS_COP.1(2)<br>FCS_COP.1.(3)<br>FPT_TUD_EXT.1 | FCS_COP.1(2) and FCS_COP.1(3) specify digital signature algorithms and hash functions used in verification of updates.<br>FPT_TUD_EXT.1 provides a way to determine the version of firmware running, initiate an update, and verify the firmware/software updates to the TOE prior to installation. |

## 4.3 Security Assurance Requirements

102 The Security Objectives for the TOE in Section 3.1 were constructed to address threats identified in Section 2.1 and the Organizational Security Policies cited in Section 2.2. The Security Functional Requirements (SFRs) in Section 4.1 are a formal instantiation of the Security Objectives.

103 As indicated in the introduction to Section 4, while this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed in Section 4.1 as well as in this section.

104 For each family, "Developer Notes" are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities (to those already contained in Section 4.1) are described as a whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in Section 4.1.

105 The TOE security assurance requirements, summarized in Table 11, identify the management and evaluative activities required to address the threats and policies identified in Section 2 of this PP. Section 4.4 provides a succinct justification for choosing this set of assurance requirements for this PP.

**Table 10: TOE Security Assurance Requirements**

| Assurance Class | Assurance Components | Assurance Components Description |
|---|---|---|
| **Development** | ADV_FSP.1 | Basic Functional Specification |
| **Guidance Documents** | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative User guidance |
| **Tests** | ATE_IND.1 | Independent testing - conformance |
| **Vulnerability Assessment** | AVA_VAN.1 | Vulnerability analysis |
| **Life Cycle Support** | ALC_CMC.1 | Labeling of the TOE |
| | ALC_CMS.1 | TOE CM coverage |

### 4.3.1 Class ADV: Development

106 For TOEs conforming to this PP, the information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. While it is not required that the TOE developer write the TSS, the TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 4.1 should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

#### 4.3.1.1 ADV_FSP.1 Basic functional specification

107 The functional specification describes the TOE Security Function Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this PP will necessarily have interfaces to the operational environment that are not directly invokable by TOE users (to include administrative users), there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. The activities for this family for this PP should focus on understanding the interfaces presented in the TSS in response to the functional requirement, and the interfaces presented in the AGD documentation. No additional "functional specification" document should be necessary to satisfy the assurance activities specified.

108 In understanding the interfaces to the TOE, it is important to consider that the threat that is to be countered is the confidentiality and integrity of the user data transmitted across the network (either via a TOE peer-to-peer connection, or TOE to VPN Gateway connection), as well as any authentication data that may traverse the connection. Additionally, the TOE, depending on its configuration, may offer protection of unauthorized access to the network behind the TOE. In addition to the network interface, the administrative interface (how the TOE is configured) also needs to be described.

109 The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

**Developer action elements:**

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.2D The developer shall provide a tracing from the functional specification to the SFRs.

Developer Note: As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPR and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

**Content and presentation elements:**

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**Evaluator action elements:**

ADV_ FSP.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ADV_ FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

***Assurance Activity:***

110 *There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.*

### 4.3.2 Class AGD: Guidance Documents

111 The guidance documents will be provided with the developer's security target. Guidance must include a description of the administrative model, and how the administrator verifies that the operational environment (the system that hosts the VPN Client) can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an administrator.

112 Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes

- instructions to successfully install the TOE in that environment; and
- instructions to manage the security of the TOE as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability through the use of either TOE capabilities, environmental capabilities, or a combination of the two.

113 Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified in Section 4.1

#### 4.3.2.1 AGD_OPE.1 Operational User Guidance

**Developer action elements:**

AGD_OPE.1.1D The developer shall provide operational user guidance.

Developer Note: Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluators will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

**Content and presentation elements:**

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

**Evaluator action elements:**

AGD_OPE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

***Assurance Activity:***

114 *During operation, the activities to be described in the guidance fall into two broad categories; those that are performed by a (non-administrative) user, and those that are performed by an administrator. It should be noted that most procedures needed for non-administrative users are referenced in the assurance activities in Section 4.1.*

115 *With respect to the administrative functions, while several have also been described in Section 4.1, additional information is required as follows.*

116 *The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.*

117 *The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.*

118 *The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:*

- *For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.*

- *Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).*

- *Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.*

#### 4.3.2.2 AGD_PRE.1 Preparative Procedures

**Developer action elements:**

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Developer Note: As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

**Content and presentation elements:**

AGD_ PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_ PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**Evaluator action elements:**

AGD_ PRE.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AGD_ PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

***Assurance Activity:***

119 *As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms and components (that is, combination of hardware and operating system) claimed for the TOE in the ST.*

120 *The evaluator shall check to ensure that the following guidance is provided:*

- *As indicated in the introductory material, administration of the TOE is performed by one or more administrators that are a subset of the group of all users of the TOE. While it must be the case that the overall system (TOE plus Operational Environment) provide this capability, the responsibility for the implementation of the functionality can vary from totally the Operational Environment's responsibility to totally the TOE's responsibility. At a high level, the guidance must contain the appropriate instructions so that the Operational Environment is configured so that it provides the portion of the capability for which it is responsible. If the TOE provides no mechanism to allow separation of administrative users from the population of users, then the instructions, for instance, would cover the OS configuration of the OS I&A mechanisms to provide a unique (OS-based) identity for users, and further guidance would instruct the installer on the configuration of the DAC mechanisms of the OS using the TOE administrative identity (or identities) so that only TOE administrators would have access to the administrative executables. If the TOE provides some or all of this functionality, then the appropriate requirements are included in the ST from Appendix C, and the assurance activities associated with those requirements provide details on the guidance necessary for both the TOE and Operational Environment.*

*The evaluators shall also perform the following tests:*

- *Test 1 [Conditional]: If the separation of administrative users from all TOE users is performed exclusively through the configuration of the Operational Environment, the evaluators will, for each configuration claimed in the ST, ensure that after configuring the system according to the administrative guidance, non-administrative users are unable to access TOE administrative functions.*

### 4.3.3 Class ATE: Tests

121 Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised

functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

#### 4.3.3.1 ATE_IND.1 Independent Testing - Conformance

122 Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operation) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 4.1 are being met, although some additional testing is specified for SARs in Section 4.3. The Assurance Activities identify the minimum testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

**Developer action elements:**

ATE_IND.1.1D The developer shall provide the TOE for testing.

**Content and presentation elements:**

ATE_IND.1.1C The TOE shall be suitable for testing.

**Evaluator action elements:**

ATE_IND.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

***Assurance Activity:***

123 *The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.*

124 *The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platform and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.*

125 *The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform.*

126 *The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.*

### 4.3.4 Class AVA: Vulnerability Assessment

127 For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, evaluators will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

#### 4.3.4.1 AVA_VAN.1 Vulnerability Survey

**Developer action elements:**

AVA_VAN.1.1D The developer shall provide the TOE for testing.

**Content and presentation elements:**

AVA_VAN.1.1C The TOE shall be suitable for testing.

**Evaluator action elements:**

AVA_VAN.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

***Assurance Activity:***

128 *As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in VPN Client products in general, as well as those that pertain to*

*the particular TOE . The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, for example, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires an electron microscope and a tank of liquid nitrogen, for instance, then a test would not be suitable and an appropriate justification would be formulated.*

### 4.3.5 Class ALC: Life-cycle Support

129 At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation at this assurance level.

#### 4.3.5.1 ALC_CMC.1 Labeling of the TOE

130 This component is targeted at identifying the TOE such that it can be distinguished from other products or version from the same vendor and can be easily specified when being procured by an end user.

**Developer action elements:**

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

**Content and presentation elements:**

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

**Evaluator action elements:**

ALC_CMC.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

***Assurance Activity:***

131 *The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.*

#### 4.3.5.2 ALC_CMS.1 TOE CM coverage

132 Given the scope of the TOE and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

**Developer action elements:**

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

**Content and presentation elements:**

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

**Evaluator action elements:**

ALC_CMS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

***Assurance Activity:***

133 *The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.*

## 4.4 Rationale for Security Assurance Requirements

134 The rationale for choosing these security assurance requirements is that this is the first Standard Protection Profile for this technology. The first Protection Profile is used to ascertain best development practices. If vulnerabilities are found in these types of products, then more stringent security assurance requirements will be mandated based on actual vendor practices.

# Appendix A: Supporting Tables, References and Acronyms

[1] Common Criteria for Information Technology Security Evaluation (CC) Version 3.1, R3 July 2009

[2] Draft Consistency Instruction Manual, for Basic Robustness Environments, Release 4.0, CC version 3.1, 2008

[3] Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology, May 25, 2001 (CHANGE NOTICES (12-03-2002)

[4] Federal Information Processing Standard Publication (FIPS-PUB) 180-3, Secure Hash Standard, October 2008

[5] Federal Information Processing Standard Publication (FIPS-PUB) 186-3, Digital Signature Standard (DSS), June 2009

[6] Federal Information Processing Standard Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001

[7] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, May 2004

[8] NIST Special Publication 800-57, Recommendation for Key Management, March 2007

[9] NIST Special Publication 800-63, Electronic Authentication Guideline, April 2006

[10] NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised) , March 2007

[11] NSA Glossary of Terms Used in Security and Intrusion Detection, Greg Stocksdale, NSA Information Systems Security Organization, April 1998. Need to update to CNSS 4009

[12] RFC 2865 Remote Authentication Dial In User Service (RADIUS), June 2000

[13] RFC 2868 RADIUS Attributes for Tunnel Protocol Support, June 2000

[14] RFC 3575 IANA Considerations for RADIUS, July 2003

[15] RFC 3579 RADIUS (Remote Authentication Dial In User Service Support For Extensible Authentication Protocol (EAP), September 2003

[16] RFC 3580 IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines, September 2003

[17] RFC 5216 The EAP-TLS Authentication Protocol, March 2008

[18] WPA2 Standard

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AF | Authorization factor |
| AS | Authorization subsystem |
| CAVS | Cryptographic Algorithm Validation System |
| CC | Common Criteria |
| CCTL | Common Criteria Testing Laboratory |
| CM | Configuration management |
| COTS | Commercial Off-The-Shelf |
| CMVP | Cryptographic Module Validation Program |
| DRBG | Deterministic Random Bit Generator |
| DoD | Department of Defense |
| EAL | Evaluation Assurance Level |
| ES | Encryption Subsystem |
| FIPS | Federal Information Processing Standards |
| ISSE | Information System Security Engineers |
| IT | Information Technology |
| OSP | Organization Security Policy |
| PP | Protection Profile |
| PUB | Publication |
| RBG | Random Bit Generator |
| SAR | Security Assurance Requirements |
| SF | Security Function |
| SFR | Security Functional Requirement |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |

# Appendix B: NIST SP 800-53/CNSS 1253 Mapping

Several of the NIST SP 800-53/CNSS 1253 controls are either fully or partially addressed by compliant TOEs. This section outlines the requirements that are addressed, and can be used by certification personnel to determine what, if any, additional testing is required when the TOE is incorporated into its operational configuration.

*Application Note: In this version, only a simple mapping is provided. In future versions, additional narrative will be included that will provide further information for the certification team. This additional information will include details regarding the SFR to control mapping discussing what degree of compliance is provided by the TOE (e.g., fully satisfies the control, partially satisfies the control). In addition, a comprehensive review of the specified assurance activities, and those evaluation activities that occur as part of satisfying the SARs will be summarized to provide the certification team information regarding how compliance was determined (e.g., document review, vendor assertion, degree of testing/verification). This information will indicate to the certification team what, if any, additional activities they need to perform to determine the degree of compliance to specified controls.*

*Since the ST will make choices as far as selections, and will be filling in assignments, a final story cannot necessarily be made until the ST is complete and evaluated. Therefore, this information should be included in the ST in addition to the PP. Additionally, there may be some necessary interpretation (e.g.,"modification") to the activities performed by the evaluator based on a specific implementation. The scheme could have the oversight personnel (e.g., Validators) fill in this type of information, or could have this done by the evaluator as part of the assurance activities. The verification activities are a critical piece of information that must be provided so the certification team can determine what, if anything, they need to do in addition to the work of the evaluation team.*

| Identifier | Name | Applicable SFRs |
|---|---|---|
| AC-3 | Access Enforcement | FMT_SMF.1 |
| AU-2 | Auditable Events | FAU_GEN.1 |
| AU-2(4) | | FAU_GEN.1 |
| AU-3 | Content of Audit Records | FAU_GEN.1 |
| AU-3(1) | | FAU_GEN.1 |
| AU-7 | Audit Reduction and Report Generation | FAU_SEL.1 |
| AU-10 | Non-Repudiation | FCS_COP.1(2) |
| AU-12 | Audit Generation | FAU_GEN.1 |
| CM-5 | Access Restrictions for Change | FPT_TUD_EXT.1 |
| IA-3 | Device Identification and Authentication | FCS_IPSEC_EXT.1, FTP_ITC.1 |
| IA-5 | Authenticator Management | FIA_PSK_EXT.1, FIA_X509_EXT.1 |
| SC-4 | Information in Shared Resources | FDP_RIP.2 |
| SC-8 | Transmission Integrity | FCS_IPSEC_EXT.1, FTP_ITC.1 |
| SC-9 | Transmission Confidentiality | FCS_IPSEC_EXT.1, FTP_ITC.1 |
| SC-12 | Cryptographic Key Establishment and Management | FCS_CKM.1, FCS_CKM_EXT.4 |
| SC-13 | Use of Cryptography | FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1 |
| SI-6 | Security Functionality Verification | FPT_TST_EXT.1 |

# Appendix C: Additional Requirements

135 *For this draft of the PP, this appendix contains additional components without supporting threats, objectives, rationale, or (in some cases) assurance activities. In tandem with the first review cycle, this supporting information will be developed and incorporated into the next release of the PP. Comments on the information contained in this section (both on whether the requirements contained are applicable to the potential conformant TOEs as well as requirements that are not contained in this appendix that are widely applicable to VPN Client products) are welcome and solicited.*

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE) are contained in the body of this PP. There are additional requirements that may be included in the ST such that the TOE is still conformant to this PP; those requirements are contained in this Appendix. This Appendix has two distinct types of requirements. Those in section C.1 are required to be implemented either by the TOE or by the Operational Environment, and deal the use of X.509 certificates. If the TOE chooses to implement this functionality rather than relying on the Operational Environment, then these requirements will be moved to the body of the ST by the ST author.

136 Requirements in sections C.2 and later, on the other hand, are **not** required, but can be implemented by the TOE. In these cases, the ST author will take the appropriate information from this Appendix and include it in their ST. Note that the ST author is responsible for ensuring that requirements that may be associated with Appendix C requirements but are not listed (e.g., FMT-type requirements) are also included in the ST. Requirements not contained in this appendix are subject to review and acceptance by the National Scheme overseeing the evaluation before a conformance claim to this PP can be made.

## C.1 Class: Identification and Authentication (FIA)

The body of the PP levies requirements on the TOE that mandates the use of X.509 certificates for the authentication of the IPsec endpoints (FIA_X509_EXT.1.1, FCS_IPSEC_EXT.1.12). The companion VPN Gateway EP levies these requirements as well. Since this is a software only product, it is possible that the TOE relies on the underlying operating system to perform some aspects of handling/managing X.509 certificates. The first of these is FIA_X509_EXT.1.2, where the TOE may offer some level of protection of the certificates, but most assuredly, the underlying OS provides the ultimate protection of the certificates from unauthorized modification or deletion. Another aspect is the determination of whether a certificate is valid - FIA_X509_EXT.1.5 or if the certificate path has a valid CA certificate FIA_X509_EXT.1.6 and FIA_X509_EXT.1.7. If this function is performed by the TOE, this requirement will be moved to the body of the PP. If the validity checking is performed by the environment and a valid or invalid response is provided to the TOE, then the requirement is left here in appendix C.1.

### X509 Certificates (FIA_X509_EXT)

#### FIA_X509_EXT.2 Extended: X.509 Certificate Storage and Management

FIA_X509_EXT.1.2 The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

*Application Note:*

137 *FIA_X509_EXT.1.2 applies to certificates that are used and processed by the TSF. Certificates that are used and process by other components in the Operational Environment (e.g., the RADIUS server) are not intended to be covered by this element.*

***Assurance Activity:***

138 *The evaluator shall ensure the TSS describes all certificate stores implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access. This description indicates whether the TOE plays any role in the protection of certificates or if it relies solely on the environment to provide the protection.*

*The evaluator shall examine the guidance documentation to ensure it describes how to configure either the TOE or the environment to prevent unauthorized modification or deletion of the certificates.*

139 *The evaluator shall perform the following tests for each function in the system that requires the use of certificates:*

*Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.*

FIA_X509_EXT.1.5 The TSF shall validate the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].

FIA_X509_EXT.1.6 The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.

FIA_X509_EXT.1.7 The TSF shall not treat a certificate as a CA certificate if the basicConstraints extension is not present or the cA flag is not set to TRUE.

*Application Note:*
*While the choice of revocation method employed in left to the ST author, future versions of this PP will mandate both methods be available to the TOE's Administrator.*

***Assurance Activity:***
*The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place – the TOE, or the environment. It may be that the TOE requests the environment to perform the check and provide a result, or the TOE may do the check itself. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.*

*The evaluator ensures the guidance documentation provides the user with the necessary information to setup the validation check whether it is done by the TOE or environment. The guidance documentation provides instructions how to select the method used for checking, as well as how to setup a protected communication path with the entity providing the information pertaining to certificate validity.*

*Test 1: 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 draft of the EP, 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 2: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.*

*Test 3: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails.*

*Test 4: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.*

## C.2 Class: Security Audit (FAU)

If audit generation is provided by the TOE, the following audit requirements must be included in the ST with the ST Author making the appropriate selections and assignments. The Threat, objective and rationale is also included here and would be moved to the Security Problem Definition presented in Section 2.

| | |
|---|---|
| T.UNDETECTED_ACTIONS | Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. |

| | |
|---|---|
| O.SYSTEM_MONITORING | The TOE will provide the capability to generate audit data. |

| | | |
|---|---|---|
| T.UNDETECTED_ACTIONS<br>Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects | O.SYSTEM_MONITORING<br>The TOE will provide the capability to generate audit data. | O.SYSTEM_MONITORING mitigates this threat by providing the administrator with the capability of configuring the audit mechanism to record actions based on a number of criteria. |

| | | |
|---|---|---|
| cannot be effectively mitigated. | | |

## Security audit data generation (FAU_GEN)

**FAU_GEN.1** **Audit Data Generation**

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All* administrative actions;
d) [*Specifically defined auditable events listed in Table 9*].

*Application Note:*

140 *The ST author can include other auditable events directly in the table; they are not limited to the list presented.*

141 *In the case of "a", the audit functions referred to are those provided by the TOE. For example, in the case that the TOE was a stand-alone executable, auditing the startup and the shutdown of the TOE itself would be sufficient to meet the requirements of this clause.*

142 *Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is present in Table 9. While the TOE itself does not need to provide the ability to perform I&A for an administrator, this requirement implies that the TOE possess the capability to audit the events described by the PP as "administrative actions" (primarily dealing with configuration of the functionality provided by the TOE). It is expected that the OPE guidance detail the steps needed to ensure the audit data generated by the TOE is integrated with the audit capabilities of the underlying IT environment.*

***Assurance Activity:***

143 *The evaluator shall check the operational guidance and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 9.*

144 *The evaluator shall in particular ensure that the operational guidance is clear in relation to the contents for failed cryptographic events. In Table 9, information detailing the cryptographic mode of operation and a name or identifier for the object being encrypted is required. The evaluator shall ensure that name or identifier is sufficient to allow an administrator reviewing the audit log to determine the context of the cryptographic operation (for example, performed during a key negotiation exchange, performed when encrypting data for transit) as well as the non-TOE endpoint of the connection for cryptographic failures relating to communications with other IT systems.*

145 *The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The TOE may contain functionality that is not evaluated in the context of this PP*

*because the functionality is not specified in an SFR. This functionality may have administrative aspects that are described in the operational guidance. Since such administrative actions will not be performed in an evaluated configuration of the TOE, the evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP, which thus form the set of "all administrative actions". The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.*

146 *The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records in accordance with the assurance activities associated with the functional requirements in this PP. Additionally, the evaluator shall test that each administrative action applicable in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.*

147 *Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.*

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of the table below*].

*Application Note:*

148 *As with the previous component, the ST author should update Table 9 with any additional information generated. "Subject identity" in the context of this requirement could either be the administrator's user id or the affected network interface, for example.*

***Assurance Activity:***

149 *This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.*

**Table 11: Auditable Events**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None. | |
| FAU_SEL.1 | All modifications to the audit configuration that occur while the audit collection functions are operating. | None. |
| FCS_CKM.1 | Failure of the key generation activity. | None. |
| FCS_CKM_EXT.4 | Failure of the key zeroization process. | Identity of object or entity being cleared. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_COP.1(1) | Failure of encryption or decryption. | Cryptographic mode of operation, name/identifier of object being encrypted/decrypted. |
| FCS_COP.1(2) | Failure of cryptographic signature. | Cryptographic mode of operation, name/identifier of object being signed/verified. |
| FCS_COP.1(3) | Failure of hashing function. | Cryptographic mode of operation, name/identifier of object being hashed. |
| FCS_COP.1(4) | Failure in Cryptographic Hashing for Non-Data Integrity. | Cryptographic mode of operation, name/identifier of object being hashed. |
| FCS_IPSEC_EXT.1 | Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.<br>Failure to establish an IPsec SA.<br>Establishment/Termination of an IPsec SA. | Presumed identity of source subject.<br>Identity of destination subject.<br>Transport layer protocol, if applicable.<br>Source subject service identifier, if applicable.<br>The entry in the SPD that applied to the decision.<br>Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. |
| FCS_RBG_EXT.1 | Failure of the randomization process. | None. |
| FDP_RIP.2 | None. | |
| FIA_PSK_EXT.1 | None. | |
| FIA_X509_EXT.1 | None. | |
| FMT_SMF.1 | None. | |
| FPT_TST_EXT.1 | Execution of this set of TSF self-tests.<br>Detected integrity violations. | For integrity violations, the TSF code file that caused the integrity violation. |
| FPT_TUD_EXT.1 | Initiation of the update.<br>Any failure to verify the integrity of the update. | No additional information. |
| FTP_ITC.1 | All attempts to establish a trusted channel.<br>Detection of modification of channel data. | Identification of the non-TOE endpoint of the channel. |

## Security Audit Event Selection (FAU_SEL)

**FAU_SEL.1** **Selective Audit**

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) **event type;**
b) **success of auditable security events;**
c) **failure of auditable security events; and**
d) **[assignment: other attributes].**

*Application Note:*

150 *The intent of this requirement is to identify all criteria that can be selected to trigger an audit event. This can be configured through an interface on the client for a user/administrator to invoke, or it could be an interface that the VPN Gateway uses to instruct the client on which events are to be audited. For the ST author, the assignment is used to list any additional criteria or "none". The auditable event types are listed in Table 9.*

***Assurance Activity:***

151 *The evaluator shall review the administrative guidance to ensure that the guidance itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The administrative guidance shall also contain instructions on how to set the pre-selection, or how the VPN Gateway will configure the client, as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced.*

152 *The evaluator shall also perform the following tests:*

- *Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.*

*Test 2 [conditional]: If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.*

153 If audit review and/or storage are supported by the TOE the following audit requirements must be included in the ST, as appropriate.

## Audit Review (FAU_SAR.1)

**FAU_SAR.1** **Audit Review**

FAU_SAR.1.1 The TSF shall provide **Authorized Administrators** with the capability to read **all audit data** from the audit records.

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the ~~user~~ **Authorized Administrators** to interpret the information.

### Restricted Audit Review (FAU_SAR.2)

**FAU_SAR.2** **Restricted Audit Review**

FAU_SAR.2.1 **Refinement:** The TSF shall prohibit all users read access to the audit records **in the audit trail, except Authorized Administrators.**

**FAU_STG_EXT.4** **Prevention of Audit Data Loss**

FAU_STG_EXT.4.1 The TSF shall provide the Authorized Administrator the capability to select one or more of the following actions:

a) prevent auditable events, except those taken by the Authorized Administrator, and
b) overwrite the oldest stored audit records

to be taken if the audit trail is full.

*Application Note:*

154 *The TOE provides the Authorized Administrator the option of preventing audit data loss by preventing auditable events from occurring. The Authorized Administrator actions under these circumstances are not required to be audited. The TOE also provides the Authorized Administrator the option of overwriting "old" audit records rather than preventing auditable events, which may protect against a denial-of-service attack.*

## C.3 Class: Identification and Authentication (FIA)

155 In the case that the TOE provides administrative capability, there are a number of requirements that can be applied to specify the capability, including remote administration, local administration, and protection of the administrative session. For this version of the PP, it is acceptable to use the administrative requirements from the VPN Gateway Protection Profile to specify such a capability for the client.

156 In the case that the TOE provides the capability to store and manage certificates used during the exchanges, the following requirement can be included in the ST.

### Pre-Shared Key Composition (FIA_PSK_EXT)

157 The TOE may support pre-shared keys for use in the IPsec protocol, and may use pre-shared keys in other protocols as well. There are two types of pre-shared keys that must be supported by the TOE, as

specified in the requirements below. The first type is referred to as “text-based pre-shared keys”, which refer to pre-shared keys that are entered by users as a string of characters from a standard character set, similar to a password. Such pre-shared keys must be conditioned so that the string of characters is transformed into a string of bits, which is then used as the key.

158 The second type is referred to as “bit-based pre-shared keys” (for lack of a standard term); this refers to keys that are either generated by the TSF on a command from the administrator, or input in "direct form" by an administrator. "Direct form" means that the input is used directly as the key, with no "conditioning" as was the case for text-based pre-shared keys. An example would be a string of hex digits that represent the bits that comprise the key.

159 The requirements below mandate that the TOE must support both text-based and bit-based pre-shared keys, although generation of the bit-based pre-shared keys may be done either by the TOE or in the operational environment.

**FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition**

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec and [selection: *no other protocols,* [assignment: other protocols that use pre-shared keys]].

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and *[selection: [assignment: other supported lengths], no other lengths]*;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”).

FIA_PSK_EXT.1.3 The TSF shall [selection, choose at least one of: condition the text-based pre-shared keys by using [selection: SHA-1, SHA-256, SHA-512, [assignment: *method of conditioning text string*]]; be able to [selection: accept, generate using the random bit generator specified in FCS_RBG_EXT.1] bit-based pre-shared keys].

*Application Note:*

160 *In the first selection, if other protocols can use pre-shared keys, they should be listed in the assignment as well; otherwise “no other protocols” should be chosen. The intent of this requirement is that all protocols will support both text-based and bit-based pre-shared keys.*

161 *For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.*

162 *In the selection for FIA_PSK_EXT.1.3, the ST author specifies the types of pre-shared keys that are supported. If “text-based pre-shared keys” is selected, the ST author fills in the method by which the text string entered by the administrator is “conditioned” into the bit string used as the key. This can be done by using one of the specified hash functions, or some other method through the assignment statement. If “bit-based pre-shared keys” is selected, the ST author specifies whether the TSF merely accepts bit-based*

*pre-shared keys, or is capable of generating them. If it generates them, the requirement specified that they must be generated using the RBG provided by the TOE.*

***Assurance Activity:***

163 *The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.*

164 *The evaluator shall examine the TSS to ensure that it identifies all protocols that allow both text-based and bit-based pre-shared keys, and states that text-based pre-shared keys of 22 characters are supported. If "text-based pre-shared keys" is selected, for each protocol identified by the requirement, the evaluator shall confirm that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by the protocol, and that this conditioning is consistent with the last selection in the FIA_PSK_EXT.1.3 requirement.*

165 *If "bit-based pre-shared keys" is selected, the evaluator shall confirm the operational guidance contains instructions for either 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 also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.*

166 *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). Note that one or more of these tests can be performed with a single test case.*

- *Test 1: The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates 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; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.*

- *Test 3 [conditional]: If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.*

- *Test 4 [conditional]: If the TOE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.*

# Appendix D: Document Conventions

167 Except for replacing United Kingdom spelling with U.S. English spelling, the notation, formatting, and conventions used in this PP are consistent with version 3.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP reader.

168 The notation, formatting, and conventions used in this PP are largely consistent with those used in version 3.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP user. The CC allows several operations to be performed on functional and assurance requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in Appendix C4 of Part 1 of the CC 3.1. Each of these operations is used in this PP.

**Refinement Convention**

169 The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by the word "Refinement" in **bold text** after the element number and the additional text in the requirement in bold text.

**Selection Convention**

170 The **selection** operation is used to select one or more options provided by the CC in stating a requirement (see appendix C.4.3 Part 1, CC 3.1). Selections that have been made by the PP authors show the selection in **bold** characters, the brackets and the word "selection" removed. Selections to be filled in by the ST author are shown in square brackets with an indication that a selection is to be made, [selection:].

**Assignment Convention**

171 The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password (see appendix C.4.2 Part 1, CC 3.1). Showing the value in **bold** characters denotes assignments that have been made by the PP authors, the brackets and the word "assignment" are removed. Assignments to be filled in by the ST author are shown in square brackets with an indication that an assignment is to be made [assignment:].

**Iteration Convention**

172 The **iteration** operation is used when a component is repeated with varying operations (see appendix C.4.1 Part 1, CC 3.1). The iteration number (iteration_number) is show in parenthesis following the component identifier.

173 The iteration operation may be performed on every component. The PP/ST author performs an iteration operation by including multiple requirements based on the same component. Each iteration of a component shall be different from all other iterations of that component, which is realized by completing assignments and selections in a different way, or by applying refinements to it in a different way.

**Extended Requirement Convention**

174 Extended requirements are permitted if the CC does not offer suitable requirements to meet the authors' needs. **Extended requirements** must be identified and are required to use the CC class/family/component model in articulating the requirements. Extended requirements will be indicated with the "EXT" inserted within the component.

**Application Notes**

175 Application notes contain additional supporting information that is considered relevant or useful for the construction of security targets for conformant TOEs, as well as general information for developers, evaluators, and ISSEs. Application notes also contain advice relating to the permitted operations of the component.

**Assurance Activities**

176 Assurance activities serve as a Common Evaluation Methodology for the functional requirements levied on the TOE to mitigate the threat. The activities include instructions for evaluators to analyze specific aspects of the TOE as documented in the TSS, thus levying implicit requirements on the ST author to include this information in the TSS section. In this version of the PP these activities are directly associated with the functional and assurance components, although future versions may move these requirements to a separate appendix or document.

# Appendix E: Glossary of Terms

**Administrator –** a user that has administrative privilege to configure the TOE in privileged mode.

**Authentication Server (AS) –** an entity designed to facilitate the authentication of an entity (user or client) that attempts to access a protected network.

**Authorized –** an entity granted access privileges to an object, system or system entity.

**Critical Security Parameter (CSP)** – security related information, e.g. secret and private cryptographic keys, and authentication data such as passwords and PINs, whose disclosure or modification can compromise the security of a cryptographic module.

**Entropy Source –** this cryptographic function provides a seed for a random number generator by accumulating the outputs from one or more noise sources. The functionality includes a measure of the minimum work required to guess a given output and tests to ensure that the noise sources are operating properly.

**FIPS-approved cryptographic function –** a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

**IT Environment –** hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.

**Operational Environment –** the environment in which the TOE is operated.

**Private Network –** a network that is protected from access by unauthorized users or entities.

**Privileged Mode –** a TOE operational mode that allows a user to perform functions that require IT Environment administrator privileges.

**Public Network –** a network that is visible to all users and entities and does not protect against unauthorized access (e.g. internet).

**Security Assurance Requirement (SAR) –** description of how assurance is to be gained that the TOE meets the SFR.

**Security Functional Requirement (SFR) –** translation of the security objectives for the TOE into a standardized language.

**Security Target (ST) –** implementation-dependent statement of security needs for a specific identified TOE.

**Target of Evaluation (TOE) –** set of software, firmware and/or hardware possibly accompanied by guidance. For this PP the TOE is the VPN Client.

**Threat Agent -** an entity that tries to harm an information system through destruction, disclosure, modification of data, and/or denial of service.

**TOE Security Functionality (TSF) –** combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs.

**TOE Summary Specification (TSS) –** a description of how the TOE satisfies all of the SFRs.

**Unauthorized User –** an entity (device or user) who has not been authorized by an authorized administrator to access the TOE or private network.

**Unprivileged Mode** – a TOE operational mode that only provides VPN client functions for the VPN Client user.

**VPN Client** – the TOE, allows remote users to use client computers to establish an encrypted IPsec tunnel across an unprotected public network to a private network

**VPN Client User –** a user operating the TOE in unprivileged mode.

**VPN Gateway** – a component that performs encryption and decryption of IP packets as they cross the boundary between a private network and a public network

# Appendix F: PP Identification

| | |
|---|---|
| Tile: | Protection Profile for IPsec Virtual Private Network (VPN) Clients |
| Version: | 1.1 |
| Sponsor: | National Information Assurance Partnership (NIAP) |
| CC Version: | Common Criteria for Information Technology Security Evaluation (CC) Version 3.1, R3 July 2009 |
| Keywords: | Authentication Server , IKE, IPsec, PKI , VPN, VPN Client, VPN |

# Appendix G: ENTROPY DOCUMENTATION AND ASSESSMENT

The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.