PP-Module for Virtual Private Network (VPN) Clients

Version: 2.1

2019-11-14

National Information Assurance Partnership

Revision History

Version	Date	Comment
2.1	2019-11-14	Initial Release

Contents

1 Introduction

1.1 Overview

The scope of this PP-Module is to describe the security functionality of a virtual private network (VPN) client in terms of [] and to define functional and assurance requirements for such products. This PP-Module is intended for use with the following Base-PPs:

- Protection Profile for General Purpose Operating Systems (OS PP)
- Protection Profile for Mobile Device Fundamentals (MD PP)
- Protection Profile for Application Software (App PP)

These Base-PPs are all valid because a VPN client may be a specific type of stand-alone software application or a built-in component of an operating system, whether desktop or mobile. Regardless of which Base-PP is claimed, the VPN client functionality defined by this PP-Module will rely on the Base-PP. Sections 5.1, 5.2, and 5.3 of this PP-Module describe the relevant functionality for each Base-PP, including specific selections, assignments, or inclusion of optional requirements that must be made as needed to support the VPN client functionality.

1.2 Terms

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

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Operational Environment	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in a ST.
Target of Evaluation (TOE)	The product under evaluation.

1.2.2 Technical Terms

Administrator	A user that has administrative privilege to configure the TOE in privileged mode.
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.
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).
Threat Agent	An entity that tries to harm an information system through destruction, disclosure, modification of data, and/or denial of service.
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.

1.3 Compliant Targets of Evaluation

The TOE defined by this PP-Module is the VPN client, a software application that runs on a physical or virtual host platform, used to establish a secure IPsec connection between that host platform and a remote system. The VPN client is intended to be located outside or inside of a private network, and establishes a secure tunnel to an IPsec peer. For the purposes of this PP-Module, IPsec peers are defined as:

- VPN gateways
- Other VPN clients
- An IPsec-capable network device (supporting IPsec for the purposes of management)

The tunnel provides confidentiality, integrity, and data authentication for information that travels across a less trusted (sometimes public) network. All VPN clients that comply with this document will support IPsec.

This PP-Module extends the OS PP when the VPN client is installed on an operating system that is discussed in that PP (e.g., Windows, Mac OS, Linux). This PP-Module extends the MD PP when the VPN client is installed on a self-contained mobile device that is bundled with an operating system (e.g. Android, iOS). This PP-Module extends the App PP when the VPN client is provided by a third party and is a standalone application that is not a bundled part of an operating system or mobile device.

As a PP-Module of any of these PPs, it is expected that the content of this PP-Module and the chosen Base-PP be appropriately combined in the context of each product-specific Security Target. This PP-Module has been specifically defined such that there should be no difficulty or ambiguity in doing so. When this PP-Module is used, conformant TOEs are obligated to implement the functionality required in the claimed Base-PP with the additional functionality defined in this PP-Module in response to the threat environment discussed subsequently herein.

1.3.1 TOE Boundary

The TOE defined by this PP-Module is purely a software solution executing on a “platform” (some sort of operating system running on hardware). Depending on the Base-PP claimed as part of the TOE, the platform may also be part of the TOE or it may be an environmental component that the TOE vendor has no control over. Regardless of whether the platform itself is within the scope of the evaluation, the VPN client itself will rely on the platform for its execution domain and 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 PP-Module contains requirements that must be met by the TOE. Depending on the Base-PP that is claimed, there may be some variation in the applicable requirements. This is because a given Base-PP may include one or more requirements that the VPN client can inherit but are not shared amongst each possible Base-PP.

This is somewhat different than other PPs, but addresses most implementations of VPN clients where some part of the functionality of the IPsec tunnel is provided by the platform. In terms of the cryptographic primitives (random bit generation, encryption/decryption, key generation, etc.) it is actually desirable that a well-tested implementation in the platform is used rather than trying to implement these functions in each client.

Requirements that can be satisfied by either the TOE or the platform are identified in Section 5 by text such as “The [selection: TSF, TOE platform] shall…” The ST author will make the appropriate selection based on where that element is implemented. It is allowable for some elements in a component to be implemented by the TOE, while other elements in that same component be implemented by the platform (requirements on the usage of X.509 certificates is an example of where this might be the case, where using the information contained in the certificates and the implementation of revocation checking may be done by the TOE, but storage and protection of the certificates may be done by the platform). Note that in the cases where this PP-Module is used to extend the OS PP or MD PP, “the TOE” includes both the VPN client and the platform. In this case, it is appropriate to indicate that the TOE satisfies this requirement. However, the ST author should make it clear, for each of these components, which are implemented by the VPN client portion of the TOE versus the platform portion.

A Supporting Document (SD) accompanies this PP-Module and contains guidance for how to evaluate the requirements defined by the PP-Module, expressed as ‘Evaluation Activities.’ Evaluation Activities will differ based on where the function that meets the requirement is implemented. In most cases, requirements implemented by the platform will require that the evaluator examine documents pertaining to the platform (generally the ST), while requirements implemented by the TOE may require examination of the TSS, examination of the Operational Guidance, and/or execution of evaluator testing. For requirements implemented by the platform there may also be requirements that the evaluator examine the interfaces used by the TOE to access these functions on the platform to ensure that the functionality being invoked to satisfy the requirements of this PP-Module is the same functionality that was evaluated.

Given the degree of coupling between a VPN client and its underlying platform, it is expected that the client will be tested on each platform claimed in the ST. In cases where the platforms are simply different versions of the same operating system (provided by the same platform vendor), an equivalency argument may be made in lieu of testing on each version. The argument would have to demonstrate that the client interacts in exactly the same way with the versions of the OS - e.g., same APIs are used with the same parameters, the network stack is modified with exactly the same kernel modules. The evaluator uses the operational guidance to configure the TOE and underlying platform.

A TOE that conforms to this PP-Module 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.

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 7296 (with mandatory support for NAT traversal as specified in section 2.23), and 4307 to authenticate and establish session keys with the VPN entities.

In order to show that the TSF implements the RFCs correctly, the evaluator shall perform Assurance Activities documented in the Supporting Document that accompanies this PP-Module. In future versions of this PP-Module, Evaluation Activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than is currently described in this publication.

The IPsec VPN client enables encryption of all information that flows between itself and its IPsec peer. The VPN client serves as an endpoint for an IPsec VPN connection and performs a number of cryptographic functions related to establishing and maintaining that connection. If the cryptography used to perform endpoint authentication, 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.

1.4 Use Cases

A VPN client allows users on the TOE platform to establish secure IPsec communications, providing confidentiality, integrity, and protection of data, across a less trusted network in order to secure data in transit. This PP-Module defines three use cases for VPN clients. A conformant TOE will implement one or more of the use cases specified below.

Regardless of the specific usage of the TOE, the focus of the Security Functional Requirements in this PPModule is on the following fundamental aspects of a VPN client:

- Authentication of the IPsec peer
- Cryptographic protection of data in transit
- Implementation of services

A VPN client can establish VPN connectivity either to a VPN gateway with traffic bound for a remote endpoint in the private network that is protected by the VPN gateway (Use Case 1), to a VPN client peer residing on a remote endpoint in the same network as the TOE (Use Case 2), and/or to a network device with IPsec capability for the purposes of managing that device (Use Case 3). In the first case, the entire IP packet is encapsulated and a new header is applied so that the gateway can route the packet to its intended destination. This is known as tunnel mode. In the latter two cases, the original IP header is preserved and only the payload is encrypted. This is known as transport mode.

Beyond the implementation differences specified by these use cases, the remaining security functionality is expected to be implemented by all VPN clients, regardless of whether it supports one or more of the use cases. Regardless of the intended use case, VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity. Authentication of IPsec peers 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.

It is assumed that the VPN client is implemented properly and contains no critical design mistakes. The VPN client relies on the system or device on which it is installed for its proper execution. 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.

[USE CASE 1] TOE to VPN Gateway

A VPN client allows users on the TOE platform to establish an encrypted IPsec tunnel across a less trusted, often unprotected public, network to a private network (see Figure 1). In this case, the TOE provides encryption/decryption of network packets as they leave/arrive the VPN client's underlying platform. 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 TOE is responsible for encrypting the packets that are intended to be received by the target on the private network and then encapsulating these packets in a way that allows the VPN gateway to securely receive them and forward them to their final destination.

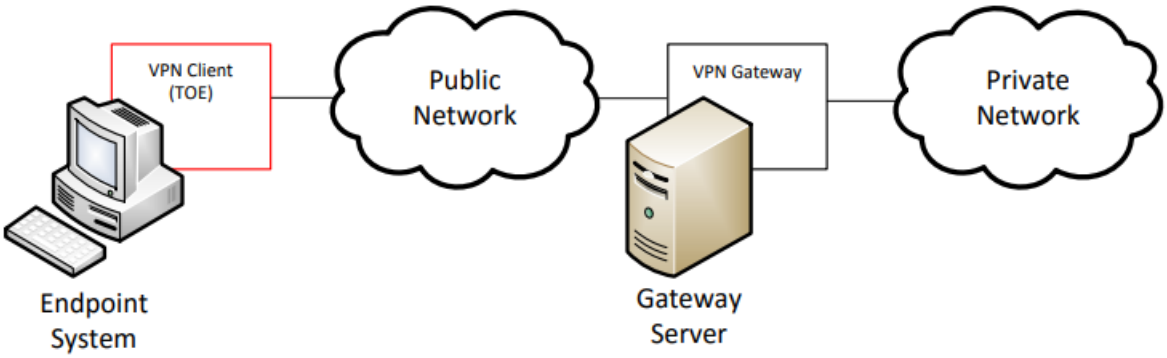

Figure 1: TOE to VPN Gateway

[USE CASE 2] TOE to VPN Client

A VPN client may additionally or alternatively allow a client computer to connect directly to another computer running a VPN client (see Figure 2). In this case, the functionality of the VPN client is to connect directly to another endpoint system in order to facilitate communications directly to that system.

IPsec transport mode is used for end-to-end communications. In this use case, the content of the packet data (payload) is encrypted but the original IP header is preserved. Inherent to this use case, when two peers are communicating directly, is the disclosure of the source/destination of the packets. Users should take into consideration any security risks associated with this disclosure when architecting their networks in line with this use case.

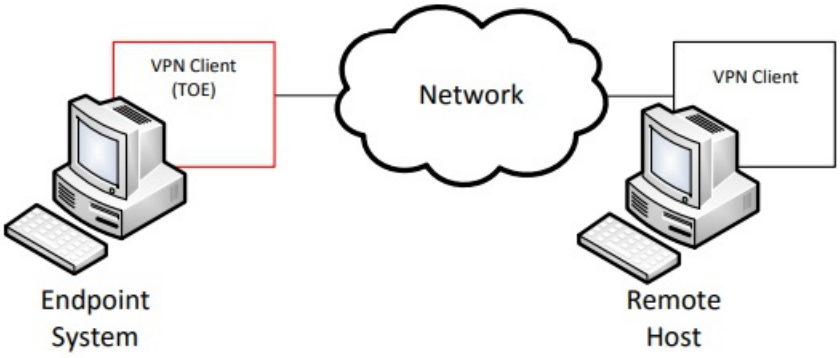

Figure 2: TOE to VPN Client

[USE CASE 3] TOE to IPsec-capable Network Device

Similar to Use Case 2 above, a VPN client TOE can also be used to establish a secure connection to an IPsec-capable network device using IPsec, similar to how SSH can be used. In this case, where a network device is being managed remotely over an IPsec connection, the network device itself must contain IPsec functionality to act as the peer for the connection (see Figure 3).

While this will behave functionally the same way as the scenario described by Use Case 2, the user of the TOE in Use Case 3 is a network administrator who is assumed to have administrative access to the network device they are connecting to.

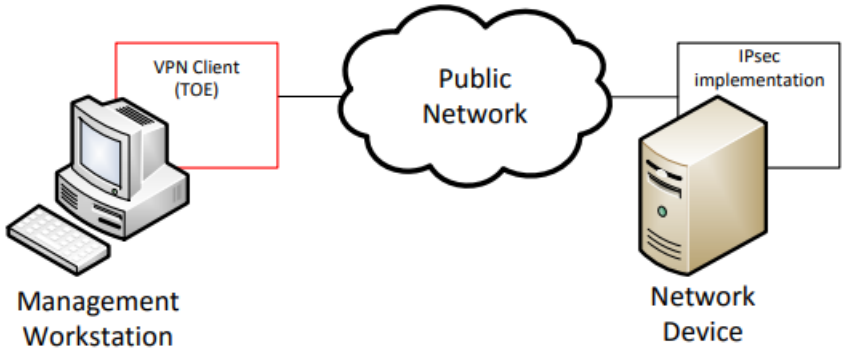

Figure 3: TOE to IPsec-capable Network Device

2 Conformance Claims

Conformance Statement

This PP-Module inherits exact conformance as required from the specified Base-PP and as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

The following PPs and PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module:

- Protection Profile for General Purpose Operating Systems, Version 4.2.1
- Protection Profile for Mobile Device Fundamentals, version 3.1
- Protection Profile for Application Software, version 1.3

CC Conformance Claims

This PP-Module is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Release 5 [CC].

Package Claims

This PP-Module does not claim conformance to any packages.

3 Security Problem Description

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about its operational environment, and any organizational security policies that the TOE is expected to enforce.

This PP-Module is written to address the situation in which a user accesses a private network (e.g. the user's office network) or terminal endpoint (e.g. a network device) using a less trusted network (such as a public Wi-Fi network or local-area network). Protection of network packets is desired as they traverse a public 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 secureVPN tunnel and performs encryption and decryption of network packets in accordance with a VPN security policy negotiated between the VPN client (TOE) and its IPsec peer.

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

Note that as a PP-Module, all threats, assumptions, and OSPs defined in the Base-PP will also apply to a TOE unless otherwise specified, depending on which of the Base-PPs it extends. The Security Functional Requirements defined in this PP-Module will mitigate the threats that are defined in the PP-Module but may also mitigate some threats defined in the Base-PPs in more comprehensive detail due to the specific capabilities provided by a VPN client.

3.1 Threats

T.UNAUTHORIZED_ACCESS

This PP-Module 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 system or device that contains the IPsec VPN client. Therefore, the primary threat agents are the unauthorized entities that try to gain access to the protected network (in cases where tunnel mode is used) or to plaintext data that traverses the public network (regardless of whether transport mode or tunnel mode is used).

The endpoint of the network communication can be both geographically and logically distant from the TOE, and can pass through a variety of other systems. These intermediate systems may be under the control of the adversary, and offer an opportunity for communications over the network to be compromised.

Plaintext communication over the network may allow critical data (such as passwords, configuration settings, and user data) to be read and/or manipulated directly by intermediate systems, leading to a compromise of the TOE or to the secured environmental system(s) that the TOE is being used to facilitate communications with. IPsec can be used to provide protection for this communication; however, there are myriad options that can be implemented for the protocol to be compliant to the protocol specification listed in the RFC. Some of these options can have negative impacts on the security of the connection. For instance, using a weak encryption algorithm (even one that is allowed by the RFC, such as DES) can allow an adversary to read and even manipulate the data on the encrypted channel, thus circumventing countermeasures in place to prevent such attacks. Further, if the protocol is implemented with little-used or non-standard options, it may be compliant with the protocol specification but will not be able to interact with other, diverse equipment that is typically found in large enterprises.

Even though the communication path is protected, there is a possibility that the IPsec peer could be duped into thinking that a malicious third-party user or system is the TOE. For instance, a middleman could intercept a connection request to the TOE, and respond to the request as if it were the TOE. In a similar manner, theTOE could also be duped into thinking that it is establishing communications with a legitimate IPsec peer when in fact it is not. An attacker could also mount a malicious man-in-the-middle-type of attack, in which an intermediate system is compromised, and the traffic is proxied, examined, and modified by this system. This attack can even be mounted via encrypted communication channels if appropriate countermeasures are not applied. These attacks are, in part, enabled by a malicious attacker capturing network traffic (for instance, an authentication session) and "playing back" that traffic in order to fool an endpoint into thinking it was communicating with a legitimate remote entity.

T.TSF_CONFIGURATION

Configuring VPN tunnels is a complex and time-consuming process, and prone to errors if the interface for doing so is not well-specified or well-behaved. The inability to configure certain aspects of the interface may also lead to the mis-specification of the desired communications policy or use of cryptography that may be desired or required for a particular site. This may result in unintended weak or plaintext communications while the user thinks that their data are being protected. Other aspects of configuring the TOE or using its security mechanisms (for example, the update process) may also result in a reduction in the trustworthiness of the VPN client.

T.UNAUTHORIZED_UPDATE

Since the most common attack vector used involves attacking unpatched versions of software containing well-known flaws, updating the VPN client is necessary to ensure that changes to threat environment are addressed. Timely application of patches ensures that the client is a "hard target," thus increasing the likelihood that product will be able to maintain and enforce its security policy. However, the updates to be applied to the product must be trustable in some manner; otherwise, an attacker can write their own "update" that instead contains malicious code of their choosing, such as a rootkit, bot, or other malware. Once this "update" is installed, the attacker then has control of the system and all of its data.

Methods of countering this threat typically involve hashes of the updates, and potentially cryptographic operations (e.g., digital signatures) on those hashes as well. However, the validity of these methods introduces additional threats. For

instance, a weak hash function could result in the attacker being able to modify the legitimate update in such a way that the hash remained unchanged. For cryptographic signature schemes, there are dependencies on

- The strength of the cryptographic algorithm used to provide the signature, and
- The ability of the end user to verify the signature, which typically involves checking a hierarchy of digital signatures back to a root of trust (a certificate authority).

If a cryptographic signature scheme is weak, then it may be compromised by an attacker and the end user will install a malicious update, thinking that it is legitimate. Similarly, if the root of trust can be compromised, then a strong digital signature algorithm will not stop the malicious update from being installed. In this case, the attacker will just create their own signature on the update using the compromised root of trust, and the malicious update will then be installed without detection.

T.USER_DATA_REUSE

Data traversing the TOE could inadvertently be sent to a different user; since these data may be sensitive, this may cause a compromise that is unacceptable. The specific threat that must be addressed concerns user data that is retained by the TOE in the course of processing network traffic that could be inadvertently re-used in sending network traffic to a user other than that intended by the sender of the original network traffic.

T.TSF_FAILURE

Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TSF.

3.2 Assumptions

These assumptions are made on the Operational Environment in order to be able to ensure that the security functionality specified in the PP-Module can be provided by the TOE. 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.

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_CONFIG

Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

3.3 Organizational Security Policies

This PP-Module defines no additional organizational security policies beyond those defined in the supported Base-PPs.

4 Security Objectives

4.1 Security Objectives for the TOE

O.PLACEHOLDER

PLACEHOLDER OBJECTIVE.

Addressed by: fcs_ipsec_ext.1

4.2 Security Objectives for the Operational Environment

The Operational Environment of the TOE implements technical and procedural measures to assist theTOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). The security objectives for the Operational Environment consist of a set of statements describing the goals that the Operational Environment should achieve. 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 3 are incorporated as security objectives for the environment. 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 section defines the security objectives that are to be addressed by the IT domain or by nontechnical or procedural means. As indicated above, if requirements supporting an objective on the TOE (in the previous table) are implemented in whole or in part by the platform, the ST should indicate this by an entry in this table with that objective.

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 theTOE and the data it contains, is assumed to be provided by the environment.

OE.TRUSTED_CONFIG

Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

4.3 Security Objectives Rationale

This section describes how the assumptions, threats, and organization security policies map to the security objectives.

Threat, Assumption, or OSP	Security Objectives	Rationale
T.UNAUTHORIZED_ACCESS	O.PLACEHOLDER	PLACEHOLDER
T.TSF_CONFIGURATION	O.PLACEHOLDER	PLACEHOLDER
T.UNAUTHORIZED_UPDATE	O.PLACEHOLDER	PLACEHOLDER
T.USER_DATA_REUSE	O.PLACEHOLDER	PLACEHOLDER
T.TSF_FAILURE	O.PLACEHOLDER	PLACEHOLDER
A.NO_TOE_BYPASS	OE.NO_TOE_BYPASS	The operational environment objective OE.NO_TOE_BYPASS is realized through A.NO_TOE_BYPASS.
A.PHYSICAL	OE.PHYSICAL	The operational environment objective OE.PHYSICAL is realized through A.PHYSICAL.
A.TRUSTED_CONFIG	OE.TRUSTED_CONFIG	The operational environment objective A.TRUSTED_CONFIG is realized through OE.TRUSTED_CONFIG.

5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following notations are used:

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

5.1 OS PP Security Functional Requirements Direction

In a PP-Configuration that includes OS PP, the TOE is expected to rely on some of the security functions implemented by the General Purpose Operating System as a whole and evaluated against the OS PP. The following sections describe any modifications that the ST author must make to the SFRs defined in the OS PP in addition to what is mandated by Section 5.4 TOE Security Functional Requirements.

5.1.1 Modified SFRs

The SFRs listed in this section are defined in the OS Protection Profile and relevant to the secure operation of the TOE.

5.1.1.1 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The OS shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm:

- **ECC schemes using "NIST curves" P-256, P-384, and [selection: *P-521*, *no other curves*] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.4,**
- **FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.1,**

[**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3* ,
- ***No other key generation methods***

].

Application Note: This SFR is functionally identical to what is defined in the OS PP except that FFC and ECC key generation have been made mandatory in support of IPsec due to the mandated support for DH groups 14, 19, and 20 in FCS_IPSEC_EXT.1.8. RSA remains present as a selection since it may be used by parts of the TSF that are not specifically related to VPN client functionality.

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The OS shall implement functionality to perform cryptographic key establishment in accordance with a specified key establishment method:

- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes using Integer Factorization Cryptography,"
- **Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography,"**

[**selection**:

- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"* ,
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3***

].

Application Note: This SFR differs from its definition in the OS PP by moving elliptic curve-based key establishment schemes from selectable to mandatory (due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8). It also provides the ability to claim either NIST SP 800-56A or RFC 3526 for key establishment using finite field cryptography. One of these two claims must be made in support of DH group 14 in FCS_IPSEC_EXT.1.8. The use of RSA is not explicitly mandated by the VPN client but it is mandatory in the OS PP as a prerequisite to implementing TLS, which is why it remains mandatory here.

FCS_COP.1/1 Cryptographic Operation (Encryption and Decryption)

FCS_COP.1.1/1 The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm

- AES-XTS (as defined in NIST SP 800-38E),
- AES-CBC (as defined in NIST SP 800-38A),
- **AES-GCM (as defined in NIST SP 800-38D),**

and [**selection**:

- *AES-CCMP (as defined in FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2012)*,
- *AES Key Wrap (KW) (as defined in NIST SP 800-38F)*,
- *AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F)*,
- *AES-CCM (as defined in NIST SP 800-38C)*,
- *AES-CCMP-256 (as defined in NIST SP800-38C and IEEE 802.11ac-2013)*,
- *AES-GCMP-256 (as defined in NIST SP800-38D and IEEE 802.11ac-2013)*,
- *no other modes*

] and cryptographic key sizes [*128-bit, 256-bit*].

Application Note: This SFR is identified in the OS PP as FCS_COP.1(1). It is otherwise identical to what is defined in the OS PP except that support for GCM mode is mandatory in order to address the requirements for FCS_IPSEC_EXT.1. In addition, both 128-bit and 256-bit for key sizes must be selected in order to meet the requirements for FCS_IPSEC_EXT.1.

5.1.2 Additional SFRs

This section defines additional SFRs that must be added to the TOE boundary in order to implement the functionality in any PP-Configuration where the OS PP is claimed as the Base-PP.

5.1.2.1 Cryptographic Support (FCS)

FCS_CKM.1/VPN Cryptographic Key Generation (IKE)

FCS_CKM.1.1/VPN The **[selection: *VPN client*, *OS*]** shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with: **[selection:**

- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3 for RSA schemes*** **,**
- ***FIPS PUB 186-4, "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]***

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

Application Note: 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 X.509v3 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.

See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

FCS_CKM_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1 The [**selection**: *VPN client*, *OS*] shall store persistent secrets and private keys when not in use in OS-provided key storage.

Application Note: This requirement ensures that persistent secrets (credentials, secret keys) and private keys are stored securely when not in use. If some secrets/keys are manipulated by the VPN client and others are manipulated by the OS, then both of the selections can be

specified by the ST author.

5.1.2.2 Identification and Authentication (FIA)

FIA_X509_EXT.3 X.509 Certificate Use and Management

FIA_X509_EXT.3.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec exchanges, and [**selection**: *digital signatures for FPT_TUD_EXT.1*, *integrity checks for FPT_TST_EXT.1*, *no additional uses*]

FIA_X509_EXT.3.2 When a connection to determine the validity of a certificate cannot be established, the [**selection**: *VPN client*, *OS*] shall [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*, *accept the certificate*, *not accept the certificate*].

Application Note: Oftentimes a connection must be established to perform a verification of the revocation status of a certificate - either to download a CRL or to perform OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). The behavior of the TOE in these cases is described by the second selection. If the TOE has determined the certificate is valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the second selection will determine the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. If the administrator-configured option is selected by the ST Author, the ST author must also make the appropriate selection in FMT_SMF.1/VPN.

FIA_X509_EXT.3.3 The [**selection**: *VPN client*, *OS*] shall not establish an SA if a certificate or certificate path is deemed invalid.

5.1.2.3 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The **[selection: *VPN client*, *OS*] shall use IPsec to provide** a **trusted** communication channel between itself and **[selection: *a remote VPN gateway*, *a remote VPN client*, *a remote IPsec-capable network device*]** 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 **[selection: *VPN client*, *OS*]** shall permit [*the TSF*] to initiate communiaction with the trusted channel.

FTP_ITC.1.3 The **[selection: *VPN client*, *OS*]** shall initiate communication via the trusted channel [*for all traffic traversing that connection*].

Application Note: The intent of the above requirement is to demonstrate that IPsec can be used to establish remote communications in transport and/or tunnel mode. 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 reestablish 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.

5.2 MD PP Security Functional Requirements Direction

In a PP-Configuration that includes MD PP, the TOE is expected to rely on some of the security functions implemented by the Mobile Device as a whole and evaluated against the MD PP. The following sections describe any modifications that the ST author must make to the SFRs defined in the MD PP in addition to what is mandated by Section 5.4 TOE Security Functional Requirements.

5.2.1 Modified SFRs

The SFRs listed in this section are defined in the MD Protection Profile and relevant to the secure operation of the TOE.

5.2.1.1 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm:

- **ECC schemes using "NIST curves" P-256, P-384, and [selection: *P-521, no other***

- **ECC schemes using "NIST curves" P-256, P-384, and [selection: *P-521, no other curves*]that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.4,**

[selection:

- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.1*,**
- ***FFC schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3, Appendix B.1***

] [**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3* ,
- *Curve25519 schemes that meet the following: RFC 7748*,
- ***No other key generation methods***

].

Application Note: This SFR is functionally identical to what is defined in the MD PP except that FFC and ECC key generation have been made mandatory in support of IPsec due to the mandated support for DH groups 14, 19, and 20 in FCS_IPSEC_EXT.1.8, and support for P-256 is made mandatory due to its use in DH group 19. Curve25519 schemes are included to satisfy FDP_DAR_EXT.2.2 in the MD PP; these schemes are not used in support of IPsec. RSA remains present as a selection since it may be used by parts of the TSF that are not specifically related to VPN client functionality.

FCS_CKM.2/1 Cryptographic Key Establishment

FCS_CKM.2.1/1

The TSF shall perform cryptographic key establishment in accordance with a specified key establishment method:

- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes using Integer Factorization Cryptography,"
- **Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography,"**

[**selection**:

- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*,
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3***

].

Application Note: This SFR is identified in the MD PP as FCS_CKM.2(1). It also differs from its definition in the MD PP by moving elliptic curve-based key establishment schemes from selectable to mandatory (due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8). It also provides the ability to claim either NIST SP 800-56A or RFC 3526 for key establishment using finite field cryptography. One of these two claims must be made in support of DH group 14 in FCS_IPSEC_EXT.1.8. The use of RSA is not explicitly mandated by the VPN client but it is mandatory in the MD PP, which is why it remains mandatory here.

FCS_COP.1/1 Cryptographic Operation

FCS_COP.1.1/1

The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm

- AES-CBC (as defined in FIPS PUB 197, NIST SP 800-38A),
- AES-CCMP (as defined in FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2012),
- **AES-GCM (as defined in NIST SP 800-38D),**

and [**selection**:

- *AES Key Wrap (KW) (as defined in NIST SP 800-38F)*,
- *AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F)*,
- *AES-CCM (as defined in NIST SP 800-38C)*,
- *AES-XTS (as defined in NIST SP 800-38E)*,
- *AES-CCMP-256 (as defined in NIST SP800-38C and IEEE 802.11ac-2013)*,
- *AES-GCMP-256 (as defined in NIST SP800-38D and IEEE 802.11ac-2013)*,
- *no other modes*

] and cryptographic key sizes 128-bit key sizes and [*256-bit key sizes*].

Application Note: This SFR is identified in the MD PP as FCS_COP.1(1). It is otherwise identical to what is defined in the MD PP except that support for GCM mode and support for

256-bit key sizes are both mandatory in order to address the requirements for FCS_IPSEC_EXT.1.

5.2.1.2 Identification and Authentication (FIA)

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and** [**selection**: *TLS*, *HTTPS*, *DTLS*, ***no other protocols***], and [**selection**: *code signing for system software updates*, *code signing for mobile applications*, *code signing for integrity verification*, ***[assignment**: other uses]*, *no additional uses*].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the revocation status of a certificate, the TSF shall [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*, *accept the certificate*, *not accept the certificate*].

Application Note: This SFR is identical to what is defined in the MD PP except that support for IPsec is mandated. Since the original SFR did not explicitly require at least one of TLS, HTTPS, or DTLS to be selected, “no other protocols” has also been added as a selection in the event that IPsec is the only protocol for which the TOE uses X.509v3 certificates for authentication.

5.2.1.3 Trusted Path/Channels (FTP)

FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1.1 The TSF shall use 802.11-2012, 802.1X, EAP-TLS, **IPsec,** and [**selection**: *TLS*, *HTTPS*, *DTLS*, ***no other protocols***] to provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

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

FTP_ITC_EXT.1.3 The TSF shall initiate communication via the trusted channel for wireless access point connections, administrative communication, configured enterprise connections, and [**selection**: *OTA updates*, *no other connections*].

Application Note: This SFR is identical to what is defined in the MD PP except that support for IPsec is mandated. Additionally, since the MD PP requires ‘at least one of’ the selected protocols which previously included IPsec, ‘no other protocols’ is now available as an option in the selection.

5.2.2 Additional SFRs

This section defines additional SFRs that must be added to the TOE boundary in order to implement the functionality in any PP-Configuration where the MD PP is claimed as the Base-PP.

5.2.2.1 Cryptographic Support (FCS)

FCS_CKM.1/VPN Cryptographic Key Generation (IKE)

FCS_CKM.1.1/VPN The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with: **[selection:**

- ***FIPS PUB 186-4, “Digital Signature Standard (DSS),” Appendix B.3 for RSA schemes*****,**
- ***FIPS PUB 186-4, “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]***

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

Application Note: 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 X.509v3 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.

See NIST Special Publication 800-57, “Recommendation for Key Management” for information about equivalent key strengths.

5.3 App PP Security Functional Requirements Direction

In a PP-Configuration that includes App PP, the TOE is expected to rely on some of the security functions implemented by the Application Software as a whole and evaluated against the App PP. The following sections describe any modifications that the ST author must make to the SFRs defined in the AppPP in addition to what is mandated by Section 5.4 TOE Security Functional Requirements.

5.3.1 Modified SFRs

The SFRs listed in this section are defined in the App Protection Profile and relevant to the secure operation of the TOE.

5.3.1.1 Cryptographic Support (FCS)

FCS_CKM.1/1 Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/1 The application shall [**selection**:

- *invoke platform-provided functionality*,
- *implement functionality*

] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm

- **[ECC schemes] using [“NIST curves” P-256, P-384, and [selection: *P-521*, *no other curves*]] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS),” Appendix B.4,**

[selection:

- ***[FFC schemes] using cryptographic key sizes of [2048-bit or greater] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS),” Appendix B.1*,**
- ***[FFC schemes] using Diffie-Hellman group 14 that meet the following:RFC 3526, Section 3,***

] [**selection**:

- *[RSA schemes] using cryptographic key sizes of [2048-bit or greater] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS), Appendix B.3"*,
- *[FFC schemes] using "safe-prime" groups that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Estbalishment Schemes using Discrete Logarithm Cryptography" and [**selection**: RFC 3526, RFC 7919]*,
- ***No other key generation methods***

].

Application Note: This SFR is identified in the AppPP as FCS_CKM.1(1). It is also selection-based in the App PP depending on the selection made in FCS_CKM_EXT.1. Because key generation services (whether implemented by the TOE or invoked from the platform) are required for IPsec, this SFR is mandatory for any TOE that claims conformance to this PP-Module.

This SFR is functionally identical to what is defined in the AppPP except that ECC key generation and at least one of FFC or DH group 14 key generation have been made mandatory in support of IPsec due to the mandated support for DH groups 14, 19, and 20 in FCS_IPSEC_EXT.1.8. RSA and "safe-prime" FFC remain present as selections since they may be used by parts of the TSF that are not specifically related to VPN client functionality.

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The application shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform cryptographic key establishment in accordance with a specified key establishment method:

- **[*Elliptic curve-based key establishment schemes*] that meets the following: [*NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”*],**

[selection:

- ***[Finite field-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”]*,**
- ***[Key establishment scheme using Diffie-Hellman group 14] that meets the following: RFC 3526, Section 3,***

] and [**selection**:

- *[RSA-based key establishment schemes] that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2*,
- *[RSA-based key establishment schemes] that meets the following: [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"]*,
- *[FFC Schemes using "safe-prime" groups]that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: RFC 3526, RFC 7919]*,
- ***No other key establishment schemes***

].

Application Note: This SFR differs from its definition in the AppPP by moving elliptic curve-based key establishment schemes from selectable to mandatory (due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8). It also provides the ability to claim either NIST SP 800-56A or RFC 3526 for key establishment using finite field cryptography. One of these two claims must be made in support of DH group 14 in FCS_IPSEC_EXT.1.8. Other selections remain conditional for TSF functionality that may exist outside the VPN Client portion of the TOE.

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The application shall [**selection**: *invoke platform-provided functionality for asymmetric key generation*, *implement asymmetric key generation*].

Application Note: This selection differs from its definition in the AppPP by removing the selection for "generate no asymmetric cryptographic keys" for this PP-Module because a VPN Client TOE will either perform its own key generation or interface with the underlying platform to provide this service, either of which causes FCS_CKM.1(1) to be claimed.

FCS_COP.1/1 Cryptographic Operation

FCS_COP.1.1/1 The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

- **AES-CBC (as defined in NIST SP 800-38A) mode,**
- **AES-GCM (as defined in NIST SP 800-38D) mode,**

and [**selection**:

- *AES-XTS (as defined in NIST SP 800-38E) mode*,
- ***no other modes***

] and cryptographic key sizes [*128-bit, 256-bit*].

Application Note: This SFR is identified in the App PP as FCS_COP.1(1). It is otherwise identical to what is defined in the App PP except that support for CBC and GCM modes and support for both 128-bit and 256-bit key sizes are all mandatory in order to address the requirements for FCS_IPSEC_EXT.1. Therefore, the relevant selections have already been completed.

5.3.1.2 Identification and Authentication (FIA)

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and** [**selection**: *HTTPS*, *TLS*, *DTLS*, ***no other protocol***].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the TSF shall [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*, *accept the certificate*, *not accept the certificate*].

Application Note: This SFR is identical to what is defined in the AppPP except that mandatory support for IPsec is added. Additionally, because this SFR is selection-based in the App PP but is mandatory for VPN client usage, the 'no other protocols' selection item has been added in the case that IPsec is the TOE's only use of certificates.

5.3.1.3 Trusted Path/Channels (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall [*encrypt all transmitted [sensitive data, data] with **[IPsec as defined in the PP-Module for VPN Client***] between itself and another trusted IT product.

Application Note: This SFR is refined from what is defined in the AppPP as the ST author is forced to select the 'encrypt all transmitted sensitive data' and 'encrypt all data' options using IPsec. The ST author may iterate the SFR if the TOE supports additional trusted channels beyond the IPsec channel used for VPN client functionality.

5.3.2 Additional SFRs

This section defines additional SFRs that must be added to the TOE boundary in order to implement the functionality in any PP-Configuration where the App PP is claimed as the Base-PP.

5.3.2.1 Cryptographic Support (FCS)

FCS_CKM.1/VPN Cryptographic Key Generation (IKE)

FCS_CKM.1.1/VPN The application shall **[selection: *invoke platform-provided functionality*, *implement functionality*]** to generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with: [**selection**:

- *FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3 for RSA schemes*,
- *FIPS PUB 186-4, "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]*

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

Application Note: 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.

See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

FCS_CKM_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1 The [**selection**: *TOE*, *TOE platform*] shall store persistent secrets and private keys when not in use in platform-provided key storage.

Application Note: This requirement ensures that persistent secrets and private keys are stored securely when not in use. This differs from FCS_STO_EXT.1 in the App PP, which only applies to secure storage of administrative credentials. If some secrets/keys are manipulated by the TOE and others are manipulated by the platform, then both of the selections can be specified by the ST author.

FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The [**selection**: *TOE*, *TOE platform*] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Application Note: 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. 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.

In practice, the TOE will not implement all of the functionality associated with the requirement, since if it performs zeroization at all it will be by invoking platform interfaces to perform the storage location clear/overwrite function. The ST author should select "TOE" when, for at least one of the keys needed to meet the requirements of this PP, the TOE manipulates (reads, writes) the data identified in the requirement and thus needs to ensure that those data are cleared. In these cases, 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.

In the likely event that some of the data are manipulated by the TOE and other data are manipulated entirely by the platform, the ST author shall select both options and make it clear in the TSS the entity responsible (TOE, TOE platform) for performing the zeroization.

5.4 TOE Security Functional Requirements

The following section describes the SFRs that must be satisfied by any TOE that claims conformance to this PP-Module. These SFRs must be claimed regardless of which PP-Configuration is used to define the TOE.

5.4.1 Cryptographic Support (FCS)

In order to show that the TSF implements the RFCs in accordance with the requirements of this PP-Module, the evaluator shall perform the evaluation activities listed in the Supporting Document. In future versions of this PP-Module, evaluation activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than are currently described in this publication. The TOE is required to use the IPsec protocol to establish connections used to communicate with an IPsec peer.

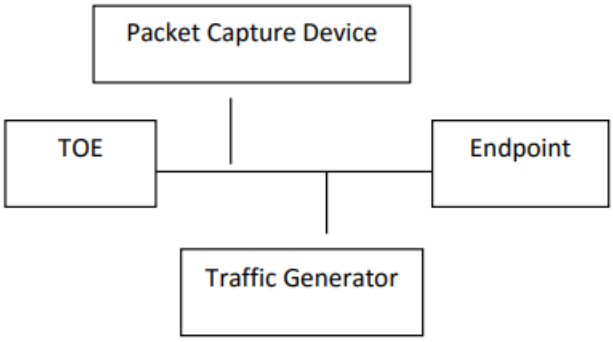

Figure 4: Sample Test Environment

The evaluator 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 evaluator must provide justification for any differences in the test environment.

In the following elements of the FCS_IPSEC_EXT.1 component, it is allowable for some or all of the individual elements to be implemented by the platform on which the VPN client operates. If the TOE extends the App PP, the ST author will indicate whether the TOE or TOE platform implements the VPN functionality through the selections in the components of this SFR. If the TOE extends the OS PP or MD PP, all IPsec functionality will be implemented by the TOE. In this case, the ST author must identify in the TSS which functions are implemented by the VPN client portion of the TOE versus the underlying operating system or mobile device.

If the configuration is to be performed on the platform, the evaluator shall ensure that the "operational guidance" for each platform in the VPN client ST contains the appropriate information (either through reference in the platform's ST, or by information contained in the VPN client ST). All tests must be performed by the evaluator using the VPN client and a representative sample of platforms listed in the VPN client ST.

FCS_IPSEC_EXT.1 IPsec

FCS_IPSEC_EXT.1.1 The [**selection**: *TOE*, *TOE platform*] shall implement the IPsec architecture as specified in RFC 4301.

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

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

A VPN gateway fully implements the IPsec capability and provides an administrative interface to establish and populate an SPD. A VPN client, on the other hand, may fully implement the IPsec functionality, or it may rely on the underlying platform to implement aspects, including the SPD. A VPN client is not required to provide an administrative interface to create or maintain an SPD. As an alternative, a client may provide an application, such as a VPN gateway, a means to establish and populate the SPD.

FCS_IPSEC_EXT.1.2 The [**selection**: *TOE*, *TOE platform*] shall implement [**selection**: *tunnel mode*, *transport mode*].

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

establish a connection to a specific endpoint device for the purpose of secure remote administration, the ST author is expected to select transport mode.

FCS_IPSEC_EXT.1.3 The [**selection**: *TOE*, *TOE platform*] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4 The [**selection**: *TOE*, *TOE platform*] 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*].

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

FCS_IPSEC_EXT.1.5 The [**selection**: *TOE*, *TOE platform*] shall implement the protocol [**selection**:

- *IKEv1, using Main Mode for Phase I exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [**selection**: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions], and [**selection**: support for XAUTH, no support for XAUTH]*,
- *IKEv2 as defined in RFC 7296 (with mandatory support for NAT traversal as specified in section 2.23), RFC 4307, and [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions]*

].

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

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

FCS_IPSEC_EXT.1.7 The [**selection**: *TOE*, *TOE platform*] shall ensure that [**selection**:

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

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

Application Note: The ST author is afforded a selection based on the version of IKE in their implementation. There is a further selection within this selection that allows the ST author to specify which entity is responsible for “configuring” the life of the SA. An implementation that allows an administrator to configure the 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; the ST author makes the appropriate selection to indicate which type of lifetime limits are supported.

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5. The IKEv1 requirement can be accomplished either by providing Authorized Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE), or by “hard coding” the limits in the implementation. For IKEv2, there are no 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 operational guidance generated for AGD_OPE. It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

FCS_IPSEC_EXT.1.8 The [**selection**: *TOE*, *TOE platform*] shall ensure that all IKE protocols implement DH groups 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), and [**selection**: *24 (2048-bit MODP with 256-bit POS)*, *15 (3072-bit MODP)*, *no other DH groups*].

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

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignments in FCS_IPSEC_EXT.1.9 and FCS_IPSEC_EXT.1.10

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

FCS_IPSEC_EXT.1.9 The [**selection**: *TOE*, *TOE platform*] shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**assignment**: *(one or more) number(s) of bits that is at least twice the “bits of security” value associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General*] bits.

FCS_IPSEC_EXT.1.10 The [**selection**: *TOE*, *TOE platform*] shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[**assignment**: *(one or more) “bits of security” value(s) associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General*].

FCS_IPSEC_EXT.1.11 The [**selection**: *TOE*, *TOE platform*] shall ensure that all IKE protocols perform peer authentication using a [**selection**: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: *Pre-shared keys*, *no other method*].

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

FCS_IPSEC_EXT.1.12 The [**selection**: *TOE*, *TOE platform*] shall not establish an SA if the [**selection**: *IP address*, *Fully Qualified Domain Name (FQDN)*, *user FQDN*, *Distinguished Name (DN)*] and [**selection**: *no other reference identifier type*, *[**assignment**: other supported reference identifier types]*] contained in a certificate does not match the expected value(s) for the entity attempting to establish a connection.

FCS_IPSEC_EXT.1.13 The [**selection**: *TOE*, *TOE platform*] shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

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

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

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

FCS_IPSEC_EXT.1.14 The [**selection**: *TOE*, *TOE platform*, *VPN Gateway*] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [**selection**: *IKEv1 Phase 1*, *IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [**selection**: *IKEv1 Phase 2*, *IKEv2 CHILD_SA*] connection.

Application Note: If "VPN Gateway" is selected, then "Test 2" in the evaluation activity may be omitted from the evaluation. Test 1, 3, and 4 shall be performed regardless of the selection.

If this functionality is configurable, the TSF may be configured by a VPN Gateway or by an

Administrator of the TOE itself.

The ST author chooses either or both of theIKE 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 this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

5.4.2 User Data Protection (FDP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The [**selection**: *TOE*, *TOE platform*] 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: 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.

5.4.3 Security Management (FMT)

The TOE is not required to maintain a separate management role. It is, however, required to provide functionality to configure certain aspects of TOE operation that should not be available to the general user population. It is possible for the TOE, TOE Platform, or VPN Gateway to provide this functionality. The client itself has to be configurable - whether it is from the EUD or from a VPN gateway.

FMT_SMF.1/VPN Specification of Management Functions (VPN)

FMT_SMF.1.1/VPN The TSF shall be capable of performing the following management functions: [**selection**:

- *Specify VPN gateways to use for connections*,
- *Specify IPsec VPN Clients to use for connections*,
- *Specify IPsec-capable network devices to use for connections*,
- *Specify client credentials to be used for connections*,
- *Configure the reference identifier of the peer*,
- *[**assignment**: any additional management functions]*

]

Application Note: Several of the management functions defined above correspond to the use cases of the TOE as follows:

- “Specify VPN gateways to use for connections” – Use Case 1
- “Specify IPsec VPN Clients to use for connections” – Use Case 2 (specifically refers to different end points to use for client-to-client connections)
- “Specify IPsec-capable network devices to use for connections” – Use Case 3

Selections appropriate for the use case(s) supported by the TOE should be claimed. "Client credentials" will include the client certificate used for IPsec authentication, and may also include a username/password.

For TOEs that support only IP address and FQDN identifier types, configuration of the reference identifier may be the same as configuration of the peer’s name for the purposes of connection.

If there are additional management functions performed by the TOE (including those specified in FCS_IPSEC_EXT.1), they should be added in the assignment.

5.4.4 Protection of the TSF (FPT)

FPT_TST_EXT.1 TSF Self-Test

FPT_TST_EXT.1.1 The [**selection**: *TOE*, *TOE platform*] 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 [**selection**: *TOE*, *TOE platform*] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [**assignment**: *cryptographic services provided either by the portion of the TOE described by the Base-PP or by the operational environment*].

Application Note: While the TOE is typically a software package running in theIT 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).

Cryptographic verification of the integrity is required, but the method by which this can be accomplished is specified in the ST in the assignment. The ST author will fill in the assignment with references to the cryptographic functions used to perform the integrity checks; this will include hashing and may potentially include digital signatures signed using X.509 certificates. If the TSF provides the cryptographic services used to verify updates, all relevant FCS_COP requirements will be identified in the assignment by the ST author.

6 Consistency Rationale

6.1 General Purpose Operating System Protection Profile

6.1.1 Consistency of TOE Type

If this PP-Module is used to extend the OS PP, the TOE type for the overall TOE is still a generalpurpose operating system. The TOE boundary is simply extended to include VPN client functionality that is built in to the operating system so that additional security functionality is claimed within the scope of the TOE.

6.1.2 Consistency of Security Problem Definition

The threats defined by this PP-Module (see section 3.1) supplement those defined in the OS PP as follows:

PP-Module Threat	Consistency Rationale
T.UNAUTHORIZED_ACCESS	The threat of an attacker gaining access to a network interface or data that is transmitted over it is consistent with the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats in the OS PP.
T.TSF_CONFIGURATION	The threat of a mis-configured VPN client is consistent with the T.LOCAL_ATTACK threat in the OS PP.
T.UNAUTHORIZED_UPDATE	The threat of an unauthorized update to theVPN client software in particular is consistent with the T.LOCAL_ATTACK threat in the OS PP.
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.NETWORK_EAVESDROP threat in the OS PP.
T.TSF_FAILURE	A failure of TSF functionality could compromise the local system, which is consistent with the T.LOCAL_ATTACK threat in the OS PP.

6.1.3 Consistency of Objectives

This PP-Module does not define any TOE objectives; the functionality provided by theVPN client assists in satisfying the O.PROTECTED_STORAGE and O.PROTECTED_COMMS in the OS PP.

The objectives for the TOEs are consistent with the OS PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.PLACEHOLDER	

The objectives for the VPN client's operational environment are consistent with theOS PP based on the rationale below.

The assumptions in both this PP-Module and the OS PP have a direct one-to-one correspondence with each other so the assumptions in this PP-Module are consistent with those in the OS PP for the same reasons as outlined in the table below.

The objectives for the TOE's Operational Environment are consistent with the OS PP based on the following rationale:

PP-Module Operational Environment Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of theOS PP and does not define an environment that an OS TOE is incapable of existing in.
OE.PHYSICAL	This is part of satisfying OE.PLATFORM as defined in theOS PP because physical security is required for hardware to be considered 'trusted.'
OE.TRUSTED_CONFIG	The expectation of trusted configuration is consistent with OE.PROPER_USER and OE.PROPER_ADMIN in the OS PP.

6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the OS PP that are needed to support Virtual Private Network (VPN) Clients functionality. This is considered to be consistent because the functionality provided by the OS is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the OS PP as well as new SFRs that are used entirely to provide functionality for Virtual Private Network (VPN) Clients. The rationale for why this does not conflict with the claims defined by the OS PP are as follows:

PP-Module Requirement	Consistency Rationale

Modified SFRs	
FCS_CKM.1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FCS_CKM.2	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FCS_COP.1/1	The SFR is refined to list an additional AES mode that must be supported to address VPN client requirements; the use of this mode for VPN connectivity does not impact the ability of the OS to satisfy any of its other security requirements.
Additional SFRs	
FCS_CKM.1/VPN	Generation of IKE peer authentication keys is added functionality that does not prevent the existing OS functions from being performed.
FCS_CKM_EXT.2	Storage of key data related to VPN functionality can be accomplished using the same mechanism defined by FCS_STO_EXT.1 in the OS PP.
FIA_X509_EXT.3	This SFR defines additional uses for X.509 certificate functionality that do not conflict with those defined in the OS PP.
FTP_ITC.1	This SFR defines a trusted channel for IPsec, which is added functionality that does not prevent the existing OS functions from being performed.
Mandatory SFRs	
FCS_IPSEC_EXT.1	This SFR defines the VPN client's IPsec implementation, which is added functionality that does not interfere with the OS functions.
FDP_RIP.2	The requirement to protect against re-use of residual data is a property of the VPN client behavior and does not impact the OS functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the OS as a whole can perform its security functions.
FPT_TST_EXT.1	Self-testing of the VPN client functionality does not impact the ability of the OS to perform its security functions.
Optional SFRs	
This PP-Module does not define any optional requirements.	
Selection-based SFRs	
FIA_PSK_EXT.1	This SFR defines the use of pre-shared keys, which is behavior that only relates to the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1	Audit records generated by the VPN client do not interfere with OS functionality. The possibility of the underlying OS platform generating audit records is consistent with the OS PP, which already contains FAU_GEN.1.
FAU_SEL.1	The ability to suppress the generation of certain audit records related to VPN activity does not interfere with the ability of the OS to satisfy its security functionality.
FDP_IFC_EXT.1	The ability of the VPN client to prevent split tunneling of IPsec traffic requires it to have hooks into lower-level OS behavior, but there are no requirements in the OS PP that would prevent this functionality from being supported.

6.2 Mobile Device Fundamentals Protection Profile

6.2.1 Consistency of TOE Type

If this PP-Module is used to extend the MD PP, the TOE type for the overall TOE is still a mobile device. The TOE boundary is simply extended to include VPN client functionality that is built in to the device's software so that additional security functionality is claimed within the scope of the TOE.

6.2.2 Consistency of Security Problem Definition

The threats defined by this PP-Module (see section 3.1) supplement those defined in the MD PP as follows:

PP-Module Threat	Consistency Rationale
T.UNAUTHORIZED_ACCESS	The threat of an attacker gaining access to a network interface or data that is transmitted over it is consistent with the T.NETWORK and T.EAVESDROP threats in the MD PP.
T.TSF_CONFIGURATION	The threat of a mis-configured VPN client is consistent with the T.NETWORK and T.EAVESDROP threats in the MD PP because failure to mitigate against mis-configuration makes these threats more significant.
T.UNAUTHORIZED_UPDATE	The threat of an unauthorized update to theVPN client software is a specific subset of the T.NETWORK threat in the MD PP.
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.EAVESDROP threat in the MD PP.
T.TSF_FAILURE	A failure of TSF functionality could compromise the local system, which is consistent with the T.FLAWAPP threat in the MD PP.

6.2.3 Consistency of Objectives

This PP-Module does not define any TOE objectives; the functionality provided by theVPN client assists in satisfying the O.COMMS, O.STORAGE, and O.CONFIG objectives in the MD PP.

The objectives for the TOEs are consistent with the MD PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.PLACEHOLDER	

The objectives for the VPN client's operational environment are consistent with theMD PP based on the rationale below.

The environmental assumptions in both this PP-Module and the MD PP have a direct one-to-one correspondence with each other so the assumptions in this PP-Module are consistent with those in the MD PP for the same reasons as outlined in the table below.

The objectives for the TOE's Operational Environment are consistent with theMD PP based on the following rationale:

PP-Module Operational Environment Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of theMD PP and does not define an environment that an MD TOE is incapable of existing in.
OE.PHYSICAL	The operational environment of a mobile device cannot guarantee physical security, but the OE.PRECAUTION objective in the MD PP ensures that an appropriate level of physical security is provided.
OE.TRUSTED_CONFIG	The expectation of trusted configuration is consistent with OE.CONFIG in the MD PP.

6.2.4 Consistency of Requirements

This PP-Module identifies several SFRs from the MD PP that are needed to support Virtual Private Network (VPN) Clients functionality. This is considered to be consistent because the functionality provided by the MD is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the MD PP as well as new SFRs that are used entirely to provide functionality for Virtual Private Network (VPN) Clients. The rationale for why this does not conflict with the claims defined by the MD PP are as follows:

PP-Module Requirement	Consistency Rationale
	Modified SFRs
FCS_CKM.1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FCS_CKM.2/1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FCS_COP.1/1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FIA_X509_EXT.2	This PP-Module adds IPsec as a new trusted protocol where x.509 certificate authentication is used.
FTP_ITC_EXT.1	This PP-Module adds IPsec as a new protocol that is used to implement trusted channels.

Additional SFRs	
FCS_CKM.1/VPN	This SFR defines the method of key generation for IKE peer authentication, which is a function that does not interfere with the functionality defined in the MD PP.
Mandatory SFRs	
FCS_IPSEC_EXT.1	This SFR defines the VPN client's IPsec implementation, which is added functionality that does not interfere with the MD functions.
FDP_RIP.2	The requirement to protect against re-use of residual data is a property of the VPN client behavior and does not impact the MD functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the MD as a whole can perform its security functions.
FPT_TST_EXT.1	Self-testing of the VPN client functionality does not impact the ability of the MD to perform its security functions.
Optional SFRs	
This PP-Module does not define any optional requirements.	
Selection-based SFRs	
FIA_PSK_EXT.1	This SFR defines the use of pre-shared keys, which is behavior that only relates to the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1	Audit records generated by the VPN client do not interfere with MD functionality. The possibility of the underlying MD platform generating audit records is consistent with the MD PP, which already contains FAU_GEN.1.
FAU_SEL.1	The ability to suppress the generation of certain audit records related to VPN activity does not interfere with the ability of the MD to satisfy its security functionality.
FDP_IFC_EXT.1	The ability of the VPN client to prevent split tunneling of IPsec traffic requires it to have hooks into lower-level mobile device behavior, but there are no requirements in the MD PP that would prevent this functionality from being supported.

6.3 Application Software Protection Profile

6.3.1 Consistency of TOE Type

If this PP-Module is used to extend the AppPP, the TOE type for the overall TOE is still a software application. The TOE boundary is made more specific by defining the TOE as a specific type of application.

6.3.2 Consistency of Security Problem Definition

The threats defined by this PP-Module (see section 3.1) supplement those defined in the App PP as follows:

PP-Module Threat	Consistency Rationale
T.UNAUTHORIZED_ACCESS	The threat of an attacker gaining access to a network interface or data that is transmitted over it is consistent with the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats in the App PP.
T.TSF_CONFIGURATION	The threat of a mis-configured VPN client is consistent with the T.LOCAL_ATTACK threat in the App PP.
T.UNAUTHORIZED_UPDATE	The threat of an unauthorized update to the VPN client software in particular is consistent with either the T.LOCAL_ATTACK or T.NETWORK_ATTACK threat in the App PP, depending on how the update is delivered.
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.NETWORK_EAVESDROP threat in the App PP.
T.TSF_FAILURE	A failure of TSF functionality could compromise the local system, which is consistent with the T.LOCAL_ATTACK threat in the App PP.

6.3.3 Consistency of Objectives

This PP-Module does not define any TOE objectives; the functionality provided by the VPN client assists in satisfying the O.PROTECTED_STORAGE and O.PROTECTED_COMMS in the App PP.

The objectives for the TOEs are consistent with the App PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.PLACEHOLDER	

The objectives for the VPN client's operational environment are consistent with the AppPP based on the rationale below.

The assumptions in both this PP-Module and the App PP have a direct one-to-one correspondence with each other so the assumptions in this PP-Module are consistent with those in the App PP for the same reasons as outlined in the table below.

The objectives for the TOE's Operational Environment are consistent with the AppPP based on the following rationale:

PP-Module Operational Environment Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of the AppPP and does not define an environment that is globally applicable to all software applications.
OE.PHYSICAL	This is part of satisfying OE.PLATFORM as defined in the AppPP because physical security is required for the underlying platform to be considered 'trustworthy.'
OE.TRUSTED_CONFIG	The expectation of trusted configuration is consistent with OE.PROPER_USER and OE.PROPER_ADMIN in the App PP.

6.3.4 Consistency of Requirements

This PP-Module identifies several SFRs from the App PP that are needed to support Virtual Private Network (VPN) Clients functionality. This is considered to be consistent because the functionality provided by the App is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the App PP as well as new SFRs that are used entirely to provide functionality for Virtual Private Network (VPN) Clients. The rationale for why this does not conflict with the claims defined by the App PP are as follows:

PP-Module Requirement	Consistency Rationale
	Modified SFRs
FCS_CKM.1/1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified. Additionally, this behavior is selection-based in the App PP but is made mandatory since it is required for VPN client functionality.
FCS_CKM.2	The ST author is instructed to make specific selections at minimum to address VPN client requirements and is modified to include Diffie-Hellman Group 14 as an additional supported method for key establishment.
FCS_CKM_EXT.1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; specifically, since key generation services are required in some capacity in order to support VPN functionality, the ST author loses the choice of stating that the application does not have any key generation functionality. Additionally, this behavior is selection-based in the App PP but is made mandatory since it is required for VPN client functionality.
FCS_COP.1/1	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FIA_X509_EXT.2	This PP-Module adds IPsec as a new trusted protocol where x.509 certificate authentication is used.
FTP_DIT_EXT.1	This PP-Module adds IPsec as a new protocol that is used to implement trusted channels.
	Additional SFRs
FCS_CKM.1/VPN	Generation of IKE peer authentication keys is added functionality that does not prevent the existing application functions from being performed.
FCS_CKM_EXT.2	This PP-Module adds a requirement for key storage, which is new functionality when compared to the Base-PP but does not interfere with its existing security functions.
FCS_CKM_EXT.4	This PP-Module adds a requirement for key destruction, which is new functionality when compared to the Base-PP but does not interfere with its existing security functions.
	Mandatory SFRs

FCS_IPSEC_EXT.1	This SFR defines the VPN client's IPsec implementation, which is added functionality that does not interfere with the application functions.
FDP_RIP.2	The requirement to protect against re-use of residual data is a property of the VPN client behavior and does not impact the general application functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the application as a whole can perform its security functions.
FPT_TST_EXT.1	Self-testing of the VPN client functionality does not impact the ability of the application to perform its security functions.
Optional SFRs	
This PP-Module does not define any optional requirements.	
Selection-based SFRs	
FIA_PSK_EXT.1	This SFR defines the use of pre-shared keys, which is behavior that only relates to the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1	Audit records generated by the VPN client do not interfere with application functionality. For cases where auditing is performed by the TOE platform, a software application is installed on a general purpose operating system or mobile device, both of which can reasonably be expected to provide audit functionality.
FAU_SEL.1	The ability to suppress the generation of certain audit records related to VPN activity does not interfere with the ability of the application to satisfy its security functionality.
FDP_IFC_EXT.1	The ability of the VPN client to prevent split tunneling of IPsec traffic requires it to have hooks into lower-level OS behavior, but there are no requirements in the App PP that would prevent this functionality from being supported.

Appendix A - Optional SFRs

This PP-Module does not define any optional SFRs.

Appendix B - Selection-based SFRs

FIA_PSK_EXT.1 Pre-Shared Key Composition

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

FIA_PSK_EXT.1.1 The [**selection**: *TOE*, *TOE platform*] shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2 The [**selection**: *TOE*, *TOE platform*] 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 [**selection**: *no other special characters*, *[**assignment**: list of additional supported special characters]*].

FIA_PSK_EXT.1.3 The [**selection**: *TOE*, *TOE platform*] shall condition the text-based pre-shared keys by using [**selection**: *SHA-1*, *SHA-256*, *SHA-512*, *[**assignment**: method of conditioning text string]*], [**selection**:

- *be able to [**selection**: accept, generate using the random bit generator specified in FCS_RBG_EXT.1]*,
- *perform no other conditioning*

].

Application Note: “TOE platform” is to be selected if the TOE claims conformance to the App PP. In the OS and MD cases, the underlying platform is still part of the TOE, so “TOE” is to be selected regardless of whether it is implemented by the VPN client itself or by the underlying operating system or mobile device.

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.

For FIA_PSK_EXT.1.3, 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 specified by the requirements. If the TOE does not use bit-based pre-shared keys, the second selection should be completed with “perform no other conditioning,” as text-based pre-shared keys would then be the only type used.

Appendix C - Objective SFRs

This section is reserved for requirements that are not currently prescribed by this PP-Module but are expected to be included in future versions of the PP-Module. Vendors planning on having evaluations performed against future products are encouraged to plan for these objective requirements to be met.

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF **and [selection: *TOE platform*, *no other component*]** 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;
c. All administrative actions;
d. [*Specifically defined auditable events listed in Auditable Events table*].

Table C-1: Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	None.
FCS_IPSEC_EXT.1	Decisions to DISCARD or BYPASS network packets processed by the TOE.	Presumed identity of source subject. Identity of destination subject. Transport layer protocol, if applicable. Source subject service identifier, if applicable. The entry in the SPD that applied to the decision.
	Failure to establish an IPsec SA.	Reason for failure. Non-TOE endpoint of connection (IP address).
	Establishment/Termination of an IPsec SA.	Non-TOE endpoint of connection (IP address).
FCS_RBG_EXT.1	Failure of the randomization process (Optional).	None.
FDP_RIP.2	None.	
FMT_SMF.1/VPN	Success or failure of management function.	None.
FPT_TUD_EXT.1	Initiation of the update.	None.
	Any failure to verify the integrity of the update.	None.

Application Note: 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.

In the case where the TOE platform is responsible for some audit functions, the ST author should identify which auditable events are generated by the TSF and which are generated by the TOE platform.

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 C-1. 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 Base-PP as "administrative actions" (primarily dealing with configuration of the functionality provided by the TOE). It is expected that the operational 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.

The auditable events listed in Table C-1 are for the SFRs that are explicitly defined in this PP-Module. This table also defines auditable events for Base-PP requirements FCS_RBG_EXT.1 and FPT_TUD_EXT.1. These are included because the App PP lacks an audit generation requirement, so if a TOE that uses the App PP as its Base-PP should also audit this App PP behavior if FAU_GEN.1 is claimed. For any SFRs that are included as part of the TOE based on the claimed Base-PP, it is expected that any applicable auditable events defined for those SFRs in the Base-PP are also claimed as part of the TSF. In other words, claiming this requirement adds to any audit requirements already present in the claimed Base-PP; it does not replace any required auditing. These auditable events only apply if the client actually performs these functions. If the platform performs any of these actions, then the platform is responsible for performing the auditing, not the TSF

FAU_GEN.1.2 The TSF **and [selection: *TOE platform*, *no other component*]** 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-Module/ST, [*information specified in column three of Auditable Events table*].

Application Note: As with the previous element, theST author should update the table with any additional information that is generated, and identify how the required information is presented by the TOE if it is ambiguous. For example, "subject identity" in the context of this requirement could refer to an administrator's user ID or to an affected network interface, depending on the event that is being generated.

FAU_SEL.1 Selective Audit

FAU_SEL.1.1 The **[selection: *TSF*, *TOE platform*]** shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes: [*event type, [success of auditable security events, failure of auditable security events], [**assignment**: list of additional attributes that audit selectivity is based upon]*].

Application Note: 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 C-1.

The intent of the first selection is to allow for the case where the underlying platform is responsible for some audit log generation functionality.

FDP_IFC_EXT.1 Subset Information Flow Control

FDP_IFC_EXT.1.1 The TSF shall ensure that all IP traffic (other than IP traffic required to establish the VPN connection) flow through the IPsec VPN client.

Application Note: This requirement is used when theVPN client is able to enforce the requirement through its own components. This generally will have to be done through using hooks provided by the platform such that the TOE is able to ensure that no IP traffic can flow through other network interfaces.

Appendix D - Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the PP-Module including those used in Appendices A through C.

D.1 Background and Scope

This appendix provides a definition for all of the extended components introduced in this PP-Module. These components are identified in the following table:

Functional Class	Functional Components
Identification and Authentication (FIA)	FIA_X509_EXT X.509 Certificate Use and Management
Cryptographic Support (FCS)	FCS_CKM_EXT Cryptographic Key Management FCS_IPSEC_EXT IPsec
Protection of the TSF (FPT)	FPT_TST_EXT TSF Self-Test
Identification and Authentication (FIA)	FIA_PSK_EXT Pre-Shared Key Composition
User Data Protection (FDP)	FDP_IFC_EXT Subset Information Flow Control

D.2 Extended Component Definitions

FIA_X509_EXT X.509 Certificate Use and Management

Family Behavior

Components in this family describe the requirements that pertain toIP traffic and information flow through the VPN client.

Component Leveling

FIA_X509_EXT.3, X.509 Certificate Use and Management, requires the TOE to perform X.509 certificate authentication and describes the behavior that is followed if is the status of the certificate is unknown or invalid.

Management: FIA_X509_EXT.3

No specific management functions are identified.

Audit: FIA_X509_EXT.3

There are no auditable events foreseen.

FIA_X509_EXT.3 X.509 Certificate Use and Management

Hierarchical to: No other components.

Dependencies to: FIA_X509_EXT.1 X.509 Certificate Validation

FPT_TST_EXT.1 TSF Self-Test

FPT_TUD_EXT.1 Trusted Update

FIA_X509_EXT.3.1

The TSF shall use X.509v3 certificates as defined byRFC 5280 to support authentication for IPsec exchanges, and [**selection**: *digital signatures for FPT_TUD_EXT.1*, *integrity checks for FPT_TST_EXT.1*, *no additional uses*]

FIA_X509_EXT.3.2

When a connection to determine the validity of a certificate cannot be established, the [**selection**: *VPN client*, *OS*] shall [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*,*accept the certificate*, *not accept the certificate*].

FIA_X509_EXT.3.3

The [**selection**: *VPN client*, *OS*] shall not establish an SA if a certificate or certificate path is deemed invalid.

FCS_CKM_EXT Cryptographic Key Management

Family Behavior
Components in this family describe requirements for key management functionality such as key storage and destruction.

Component Leveling
FCS_CKM_EXT.2, Cryptographic Key Storage,

Management: FCS_CKM_EXT.2
There are no management functions foreseen.

Audit: FCS_CKM_EXT.2
There are no audit events foreseen.

FCS_CKM_EXT.2 Cryptographic Key Storage
Hierarchical to: No other components.

Dependencies to: No dependencies.

FCS_CKM_EXT.2.1

The [**selection**: *VPN client*, *OS*] shall store persistent secrets and private keys when not in use in OS-provided key storage.

Component Leveling
FCS_CKM_EXT.4, Cryptographic Key Destruction, requires the TSF to destroy key data when no longer required.

Management: FCS_CKM_EXT.4
No specific management functions are identified.

Audit: FCS_CKM_EXT.4
There are no auditable events foreseen.

FCS_CKM_EXT.4 Cryptographic Key Destruction
Hierarchical to: No other components.

Dependencies to: No dependencies

FCS_CKM_EXT.4.1

The [**selection**: *TOE*, *TOE platform*] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

FCS_IPSEC_EXT IPsec

Family Behavior
Components in this family describe requirements for IPsec implementation.

Component Leveling
FCS_IPSEC_EXT.1, IPsec, requires the TSF to securely implement the IPsec protocol.

Management: FCS_IPSEC_EXT.1
The following actions could be considered for the management functions in FMT:

- Specify VPN gateways to use for connections
- Specify IPsec VPN Clients to use for connections
- Specify IPsec-capable network devices to use for connections
- Specify client credentials to be used for connections

Audit: FCS_IPSEC_EXT.1
The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Decisions to DISCARD or BYPASS network packets processed by theTOE
- Failure to establish an IPsec SA
- Establishment/Termination of an IPsec SA

FCS_IPSEC_EXT.1 IPsec
Hierarchical to: No other components.

Dependencies to: FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Distribution

FCS_COP.1 Cryptographic Operation

FCS_IPSEC_EXT.1.1

The [**selection**: *TOE*, *TOE platform*] shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2

The [**selection**: *TOE*, *TOE platform*] shall implement [**selection**: *tunnel mode*, *transport mode*].

FCS_IPSEC_EXT.1.3

The [**selection**: *TOE*, *TOE platform*] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4

The [**selection**: *TOE*, *TOE platform*] 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*].

FCS_IPSEC_EXT.1.5

The [**selection**: *TOE*, *TOE platform*] shall implement the protocol [**selection**:

- *IKEv1, using Main Mode for Phase I exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [**selection**: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions], and [**selection**: support for XAUTH, no support for XAUTH]*,
- *IKEv2 as defined in RFC 7296 (with mandatory support for NAT traversal as specified in section 2.23), RFC 4307, and [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions]*

].

FCS_IPSEC_EXT.1.6

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

FCS_IPSEC_EXT.1.7

The [**selection**: *TOE*, *TOE platform*] shall ensure that [**selection**:

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

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

FCS_IPSEC_EXT.1.8

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

FCS_IPSEC_EXT.1.9

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

FCS_IPSEC_EXT.1.10

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

FCS_IPSEC_EXT.1.11

The [**selection**: *TOE*, *TOE platform*] shall ensure that all IKE protocols perform peer authentication using a [**selection**: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: *Pre-shared keys*, *no other method*].

FCS_IPSEC_EXT.1.12

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

FCS_IPSEC_EXT.1.13

The [**selection**: *TOE*, *TOE platform*] shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

FCS_IPSEC_EXT.1.14

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

FPT_TST_EXT TSF Self-Test

Family Behavior

Components in this family describe requirements for self-test to verify functionality and integrity of the TOE.

Component Leveling

FPT_TST_EXT.1, TSF Self-Test, requires the TOE to perform power on self-tests to verify its functionality and the integrity of its stored executable code.

Management: FPT_TST_EXT.1

No specific management functions are identified.

Audit: FPT_TST_EXT.1

There are no auditable events foreseen.

FPT_TST_EXT.1 TSF Self-Test

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

FPT_TST_EXT.1.1

The [**selection**: *TOE*, *TOE platform*] 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 [**selection**: *TOE*, *TOE platform*] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [**assignment**: *cryptographic services provided either by the portion of the TOE described by the Base-PP or by the operational environment*].

FIA_PSK_EXT Pre-Shared Key Composition

Family Behavior

Components in this family describes the requirements for pre-shared keys when implementing IPsec

Component Leveling

FIA_PSK_EXT.1, Pre-Shared Key Composition, defines the use and composition of pre-shared keys used for IPsec

Management: FIA_PSK_EXT.1

No specific management functions are identified.

Audit: FIA_PSK_EXT.1

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Failure of the randomization process

FIA_PSK_EXT.1 Pre-Shared Key Composition

Hierarchical to: No other components.

Dependencies to: FCS_RBG_EXT.1 Random Bit Generation

FIA_PSK_EXT.1.1

The [**selection**: *TOE*, *TOE platform*] shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2

The [**selection**: *TOE*, *TOE platform*] 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 [**selection**: *no other special characters*, ***[assignment**: list of additional supported special characters]*].

FIA_PSK_EXT.1.3

The [**selection**: *TOE*, *TOE platform*] shall condition the text-based pre-shared keys by using [**selection**: *SHA-1*, *SHA-256*, *SHA-512*, ***[assignment**: method of conditioning text string]*], [**selection**:

- *be able to [**selection**: accept, generate using the random bit generator specified in FCS_RBG_EXT.1]*,
- *perform no other conditioning*

].

FDP_IFC_EXT Subset Information Flow Control

Family Behavior

Components in this family describe the requirements that pertain toIP traffic and information flow through the VPN client.

Component Leveling

FDP_IFC_EXT.1, Subset Information Flow Control, requires the TSF to process all IP traffic through its VPN client functionality.

Management: FDP_IFC_EXT.1

No specific management functions are identified.

Audit: FDP_IFC_EXT.1

There are no auditable events foreseen.

FDP_IFC_EXT.1 Subset Information Flow Control

Hierarchical to: No other components.

Dependencies to: No dependencies

FDP_IFC_EXT.1.1

The TSF shall ensure that all IP traffic (other than IP traffic required to establish the VPN connection) flow through the IPsec VPN client.

Appendix E - Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1, Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1, Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1, Revision 5, April 2017.
[OS PP]	Protection Profile for General Purpose Operating Systems, Version 4.2.1, April 2019
[MD PP]	Protection Profile for Mobile Device Fundamentals, Version 3.1, June 2017
[App PP]	Protection Profile for Application Software, Version 1.3, March 2019
[SD]	Supporting Document Mandatory Technical Document, PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, November 2019

Appendix F - Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
CC	Common Criteria
CEM	Common Evaluation Methodology
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DH	Diffie-Hellman
DN	Distinguished Name
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ESP	Encapsulating Security Protocol
EUD	End-User Device
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
FQDN	Fully Qualified Domain Name
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IT	Information Technology
MD	Mobile Device (Fundamentals)
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OS	(General Purpose) Operating System
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
PP	Protection Profile
PP-Module	Protection Profile Module
PUB	Publication
RBG	Random Bit Generation
RFC	Request For Comment
SA	Security Association
SAR	Security Assurance Requirement
SD	Supporting Document
SFR	Security Functional Requirement

SHA	Secure Hash Algorithm
SPD	Security Policy Database
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
UDP	User Datagram Protocol
VPN	Virtual Private Network