PP-Module for VPN Clients

Version: 2.4
2022-03-31
National Information Assurance Partnership

Revision History

Version	Date	Comment
2.4	2022-03-31	Incorporation of TC feedback
2.3	2021-08-10	Support for MDF, Bluetooth updates
2.2	2021-01-05	Update release
2.1	2019-11-14	Initial Release

Contents

1 Introduction
1.1 Overview
1.2 Terms
1.2.1 Common Criteria Terms
1.2.2 Technical Terms
1.3 Compliant Targets of Evaluation
1.3.1 TOE Boundary
1.4 Use Cases
1.5 Requirements Focus
2 Conformance Claims
3 Security Problem Description
3.1 Threats
3.2 Assumptions
3.3 Organizational Security Policies
4 Security Objectives
4.1 Security Objectives for the TOE
4.2 Security Objectives for the Operational Environment
4.3 Security Objectives Rationale
5 Security Requirements
5.1 VPN Clients PP Security Functional Requirements Direction
5.1.1 Modified SFRs
5.1.1.1 Cryptographic Support (FCS)
5.1.2 Additional SFRs
5.1.2.1 Cryptographic Support (FCS)
5.1.2.2 Identification and Authentication (FIA)
5.1.2.3 Trusted Path/Channels (FTP)
5.2 VPN Clients PP Security Functional Requirements Direction
5.2.1 Modified SFRs
5.2.1.1 Cryptographic Support (FCS)
5.2.1.2 User Data Protection (FDP)
5.2.1.3 Identification and Authentication (FIA)
5.2.1.4 Security Management (FMT)
5.2.1.5 Trusted Path/Channels (FTP)
5.2.2 Additional SFRs
5.3 App PP Security Functional Requirements Direction
5.3.1 Modified SFRs
5.3.1.1 Cryptographic Support (FCS)
5.3.1.2 Identification and Authentication (FIA)
5.3.1.3 Trusted Path/Channels (FTP)
5.3.2 Additional SFRs
5.3.2.1 Cryptographic Support (FCS)
5.4 MDM PP Security Functional Requirements Direction
5.4.1 Modified SFRs
5.4.1.1 Cryptographic Support (FCS)
5.4.1.2 Identification and Authentication (FIA)
5.4.1.3 Protection of the TSF (FPT)
5.4.1.4 Trusted Path/Channels (FTP)
5.4.2 Additional SFRs
5.5 TOE Security Functional Requirements
5.5.1 Auditable Events for Mandatory SFRs
5.5.2 Cryptographic Support (FCS)
5.5.3 User Data Protection (FDP)
5.5.4 Security Management (FMT)
5.5.5 Protection of the TSF (FPT)
5.6 TOE Security Functional Requirements Rationale
6 Consistency Rationale
6.1 Protection Profile for VPN Clients
6.1.1 Consistency of TOE Type

6.1.2 Consistency of Security Problem Definition
6.1.3 Consistency of Objectives
6.1.4 Consistency of Requirements
6.2 Protection Profile for VPN Clients
6.2.1 Consistency of TOE Type
6.2.2 Consistency of Security Problem Definition
6.2.3 Consistency of Objectives
6.2.4 Consistency of Requirements
6.3 Protection Profile for VPN Clients
6.3.1 Consistency of TOE Type
6.3.2 Consistency of Security Problem Definition
6.3.3 Consistency of Objectives
6.3.4 Consistency of Requirements
6.4 Protection Profile for VPN Clients
6.4.1 Consistency of TOE Type
6.4.2 Consistency of Security Problem Definition
6.4.3 Consistency of Objectives
6.4.4 Consistency of Requirements
Appendix A - Optional SFRs
A.1 Strictly Optional Requirements
A.1.1 Auditable Events for Strictly Optional SFRs
A.1.2 Identification and Authentication (FIA)
A.1.3 Packet Filtering (FPF)
A.2 Objective Requirements
A.2.1 Auditable Events for Objective SFRs
A.2.2 Security Audit (FAU)
A.3 Implementation-based Requirements
A.3.1 Auditable Events for Implementation-Dependent SFRs
A.3.2 User Data Protection (FDP)
Appendix B - Selection-based Requirements
B.1 Auditable Events for Selection-based SFRs
B.2 Cryptographic Support (FCS)
B.3 Identification and Authentication (FIA)
Appendix C - Extended Component Definitions
C.1 Extended Components Table
C.2 Extended Component Definitions
C.2.1 Cryptographic Support (FCS)
C.2.1.1 FCS_CKM_EXT Cryptographic Key Management
C.2.1.2 FCS_IPSEC_EXT IPsec
C.2.1.3 FCS_EAP_EXT EAP-TLS
C.2.2 Identification and Authentication (FIA)
C.2.2.1 FIA_X509_EXT X.509 Certificate Use and Management
C.2.2.2 FIA_BMA_EXT Biometric Activation
C.2.2.3 FIA_HOTP_EXT HMAC-Based One-Time Password Pre-Shared Keys
C.2.2.4 FIA_PSK_EXT Pre-Shared Key Composition
C.2.2.5 FIA_TOTP_EXT Time-Based One-Time Password Pre-Shared Keys
C.2.3 Packet Filtering (FPF)
C.2.3.1 FPF_MFA_EXT Multifactor Authentication Filtering
C.2.4 Protection of the TSF (FPT)
C.2.4.1 FPT_TST_EXT TSF Self-Test
C.2.5 User Data Protection (FDP)
C.2.5.1 FDP_VPN_EXT Subset Information Flow Control
Appendix D - Implicitly Satisfied Requirements
Appendix E - Entropy Documentation and Assessment
Appendix F - Acronyms
Appendix G - Bibliography

1 Introduction

1.1 Overview

The scope of this Protection Profile Module (PP-Module) is to describe the security functionality of a virtual private network (VPN) client in terms of [CC] 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 (GPOS PP), Version 4.2.1
- Protection Profile for Mobile Device Fundamentals (MDF PP), Version 3.2
- Protection Profile for Application Software (App PP), Version 1.4
- Protection Profile for Mobile Device Management (MDM PP), Version 4.0

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

(PP-Module)	
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.

1.2.2 Technical Terms

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.
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, 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 GPOS PP when the VPN client is installed on an OS discussed in that PP (e.g., Windows, Mac OS, Linux). This PP-Module extends the MDF PP when the VPN client is installed on a self-contained mobile device that is bundled with an OS (e.g. Android, BlackBerry OS, iOS, Windows Mobile). 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 OS or mobile device. This PP-Module extends the MDM PP when the VPN Client is included with MDM Server software that is used for centralized deployment and administration of enterprise mobile device policies.

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 ST. 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 in this PP-Module.

1.3.1 TOE Boundary

The TOE defined by this PP-Module is purely a software solution executing on a platform (some sort of OS 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 (OE) 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 between 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 and 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 GPOS PP or MDF 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 (EAs). EAs 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, or execution of evaluator testing. For requirements implemented by the platform, there may also be requirements where the evaluator must examine the interfaces used by the TOE to access these functions on the platform. This ensures 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 OS (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 (i.e., the 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 the Internet Key Exchange (IKE)v1 protocol, IKEv2, or both. IKEv1 is implemented as defined in RFCs 2407, 2408, 2409, 4109, and IKEv2 is implemented as specified in RFC 7296 and 4307 to authenticate and establish session keys with the VPN entities. The IKEv2 implementation also requires mandatory support for network address translation (NAT) traversal as specified in section 2.23 of RFC 7296.

In order to show that the TSF implements the RFCs correctly, the evaluator will perform EAs documented in the SD that accompanies this PP-Module. In future versions of this PP-Module, EAs may be modified or new ones may be 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:

[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 and decryption of network packets as they leave and arrive on 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.

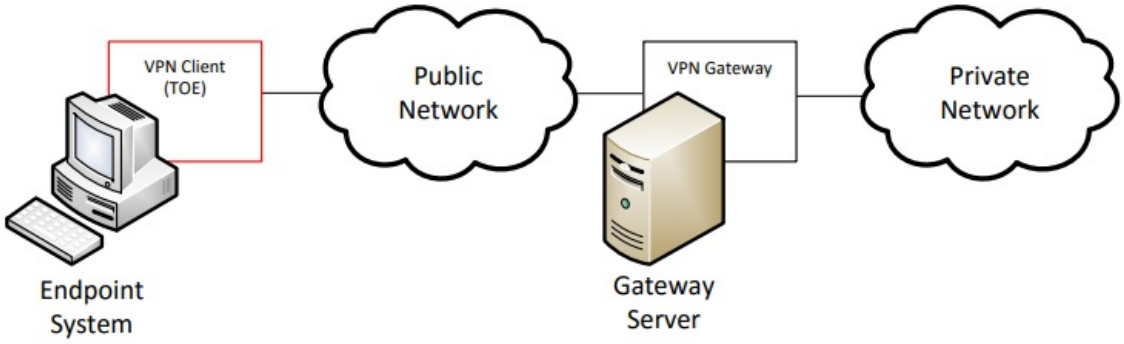


Figure 1: TOE to VPN Gateway

[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 and 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.

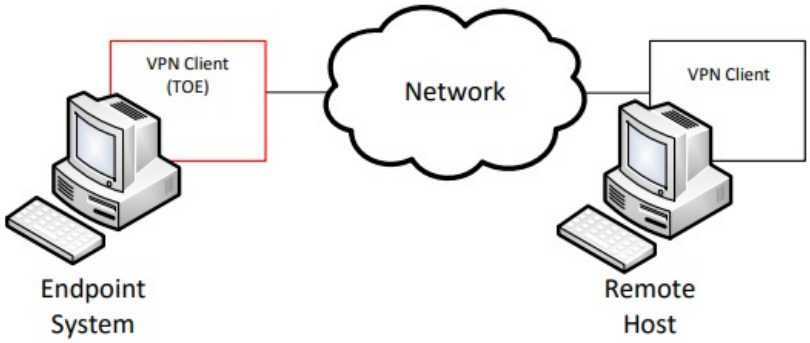


Figure 2: TOE to VPN Client

[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.

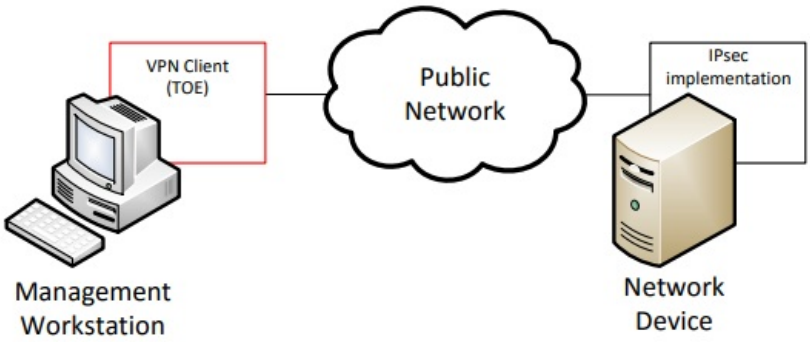


Figure 3: TOE to IPsec-capable Network Device

1.5 Requirements Focus

Regardless of the specific usage of the TOE, the focus of the Security Functional Requirements (SFRs) in this PP-Module 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), 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 OE supported.

2 Conformance Claims

Conformance Statement

This PP-Module inherits exact conformance as required from the specified Base-PPs 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:

- PP-Module for MDM Agents, Version 1.0
- PP-Module for File Encryption Enterprise Management, Version 1.0
- PP-Module for File Encryption, Version 2.0
- PP-Module for Bluetooth, Version 1.0
- PP-Module for Wireless Local Area Network (WLAN) Client, Version 1.0

CC Conformance Claims

This PP-Module is conformant to Parts 2 (extended) and 3 (extended) of Common Criteria Version 3.1, Revision 5 [CC] when App PP, GPOS PP, or MDF PP is the Base-PP.

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

PP Claim

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

Package Claim

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 OE, 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 secure VPN 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 organizational security policies (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 SFRs 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

The following threats defined in this PP-Module extend the threats defined by the Base-PPs.

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 or manipulated directly by a malicious user or process on intermediate systems, leading to a compromise of the TOE or to the secured environmental systems that the TOE is being used to facilitate communications with. IPsec can be used to provide protection for this communication; however, there are numerous 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 tricked 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, the TOE could also be tricked 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 or failure of an ignorant or careless administrator 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.USER_DATA_REUSE

Data traversing the TOE could inadvertently be sent to a different user as a consequence of a poorly-designed TOE; 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 (OE) 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 OE 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 OE will follow the applicable security configuration guidance.

3.3 Organizational Security Policies

An organization deploying the TOE is expected to satisfy the organizational security policy listed below in addition to all organizational security policies defined by the claimed Base-PP.

This document does not define any additional OSPs.

4 Security Objectives

4.1 Security Objectives for the TOE

O.AUTHENTICATION

To address the issues associated with unauthorized disclosure of information in transit, a compliant TOE's authentication ability (IPsec) will allow the TSF to establish VPN connectivity with a remote VPN gateway or peer and ensure that any such connection attempt is both authenticated and authorized.

Addressed by: FIA_X509_EXT.3 (when GPOS PP is Base-PP), FDP_IFC_EXT.1 (refined from MDF PP), FIA_X509_EXT.2 (refined from MDF PP), FIA_X509_EXT.2 (refined from App PP), FIA_X509_EXT.2 (refined from MDM PP), FCS_IPSEC_EXT.1, FIA_BMA_EXT.1 (optional), FPF_MFA_EXT.1 (optional), FCS_EAP_EXT.1 (selection-based), FIA_HOTP_EXT.1 (selection-based), FIA_PSK_EXT.1 (selection-based), FIA_PSK_EXT.2 (selection-based), FIA_PSK_EXT.3 (selection-based), FIA_PSK_EXT.4 (selection-based), FIA_PSK_EXT.5 (selection-based), FIA_TOTP_EXT.1 (selection-based).

O.CRYPTOGRAPHIC_FUNCTIONS

To address the issues associated with unauthorized disclosure of information in transit, a compliant TOE will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.

Addressed by: FCS_CKM.1 (refined from GPOS PP), FCS_CKM.2 (refined from GPOS PP) FCS_COP.1/1 (refined from GPOS PP), FTP_ITC.1 (when GPOS PP is Base-PP) FCS_CKM.1 (refined from MDF PP), FCS_CKM.2/UNLOCKED (refined from MDF PP) FCS_COP.1/ENCRYPT (refined from MDF PP), FTP_ITC_EXT.1 (refined from MDF PP) FCS_CKM.1/AK (refined from App PP), FCS_CKM.2 (refined from App PP) FCS_CKM_EXT.1 (refined from App PP), FCS_COP.1/SKC (refined from App PP), FTP_DIT_EXT.1 (refined from App PP), FCS_CKM.1 (refined from MDM PP), FCS_CKM.2 (refined from MDM PP) FCS_COP.1/1 (refined from MDM PP), FPT_ITT.1/1 (if applicable, refined from MDM PP) FTP_ITC.1/1 (if applicable, refined from MDM PP), FTP_TRP.1/1 (if applicable, refined from MDM PP) FCS_CKM.1/VPN, FCS_IPSEC_EXT.1, FCS_EAP_EXT.1 (selection-based).

O.KNOWN_STATE

The TOE will provide sufficient measures to ensure it is operating in a known state. At minimum this includes management functionality to allow the security functionality to be configured and self-test functionality that allows it to assert its own integrity. It may also include auditing functionality that can be used to determine the operational behavior of the TOE.

Addressed by: FMT_SMF_EXT.1 (refined from MDF PP), FMT_SMF.1/VPN, FPT_TST_EXT.1/VPN, FAU_GEN.1/VPN (optional), FAU_SEL.1/VPN (optional).

O.NONDISCLOSURE

To address the issues associated with unauthorized disclosure of information at rest, a compliant TOE will ensure that non-persistent data is purged when no longer needed. The TSF may also implement measures to protect against the disclosure of stored cryptographic keys and data through implementation of protected storage and secure erasure methods. The TOE may optionally also enforce split-tunneling prevention to ensure that data in transit cannot be disclosed inadvertently outside of the IPsec tunnel and prohibit transmission of packets through a connection until certain conditions are met.

Addressed by: FCS_CKM_EXT.2 (when GPOS PP is Base-PP), FCS_CKM_EXT.2 (when App PP is Base-PP), FCS_CKM_EXT.4 (when App PP is Base-PP), FDP_RIP.2, FDP_VPN_EXT.1 (optional), FPF_MFA_EXT.1 (optional).

4.2 Security Objectives for the Operational Environment

The OE 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). The security objectives for the OE consist of a set of statements describing the goals that the OE 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.

OE.NO_TOE_BYPASS

Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

OE.TRUSTED_CONFIG

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

4.3 Security Objectives Rationale

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

Table 1: Security Objectives Rationale

Threat, Assumption, or OSP	Security Objectives	Rationale
T.UNAUTHORIZED_ACCESS	O.AUTHENTICATION	The TOE mitigates the threat of unauthorized access by requiring IPsec communications to be properly authenticated.
	O.CRYPTOGRAPHIC_FUNCTIONS	The TOE mitigates the threat of unauthorized access by implementing IPsec using strong cryptographic algorithms.
T.TSF_CONFIGURATION	O.KNOWN_STATE	The TOE mitigates the threat of inadequate configuration by providing a management interface that allows all security-relevant functionality to be configured.
	OE.TRUSTED_CONFIG	This objective mitigates the threat of misconfiguration by ensuring that a malicious actor is not given direct administrative control over the TOE.
T.USER_DATA_REUSE	O.NONDISCLOSURE	The TOE mitigates the threat of data reuse by ensuring that persistently stored data is protected from unauthorized access, non-persistently stored data is appropriately purged, and potentially to ensure that no network traffic is inadvertently transmitted outside of the IPsec tunnel.
T.TSF_FAILURE	O.KNOWN_STATE	The TOE mitigates the threat of TSF failure by enforcing the use of self-tests so that the TOE remains in a known state, and potentially to generate audit records that allow for potential failures to be diagnosed.
A.NO_TOE_BYPASS	OE.NO_TOE_BYPASS	This assumption is satisfied by the environmental objective that ensures network routes do not exist that allow traffic to be transmitted from the TOE system to its intended destination without going through the TOE’s IPsec tunnel.
A.PHYSICAL	OE.PHYSICAL	This assumption is satisfied by the environmental objective that ensures the TOE is not deployed on a system that is vulnerable to loss of physical custody.
A.TRUSTED_CONFIG	OE.TRUSTED_CONFIG	This assumption is satisfied by the environmental objective that ensures that anyone responsible for administering the TOE can be trusted not to misconfigure it, whether intentionally or not.

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 conventions are used for the completion of operations:

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

5.1 VPN Clients PP Security Functional Requirements Direction

In a PP-Configuration that includes the GPOS PP, the VPN client is expected to rely on some of the security functions implemented by the OS as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.5.

5.1.1 Modified SFRs

The SFRs listed in this section are defined in the VPN Clients PP 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, and,***

[**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 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*
- *FFC Schemes using safe primes that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes*
- ***No other key generation methods***

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

Application Note: This SFR is functionally identical to what is defined in the GPOS PP except that ECC key generation with support for P-256 and P-384 has been made mandatory in support of IPsec due to the mandated support for Diffie-Hellman (DH) groups 19 and 20 in FCS_IPSEC_EXT.1.8. The ST author must select all key generation schemes used for key establishment and entity authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2 and selected cryptographic protocols must match the selection. When key generation is used for entity authentication, the public key is expected to be associated with an X.509v3 certificate.

If the OS acts only as a receiver in the RSA key establishment scheme, the OS does not need to implement RSA key generation.

Evaluation Activities

FCS_CKM.1
Refer to the evaluation activity for FCS_CKM.1 in the GPOS PP for evaluating this SFR.

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:

- ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography," and***

[**selection**:

- *RSA-based key establishment schemes that meets the following: RSAESPKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2*
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "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*
- ***No other key establishment schemes***

] ~~that meets the following [assignment: list of standards]~~.

Application Note: This SFR is functionally identical to what is defined in the GPOS PP except that elliptic curve cryptography (ECC) key generation with support for P-256 and P-384 has been made mandatory in support of IPsec due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8.

The ST author must select all key establishment schemes used for the selected cryptographic protocols.

The elliptic curves used for the key establishment scheme must correlate with the curves specified in FCS_CKM.1.1. The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

Evaluation Activities

FCS_CKM.2
Refer to the Assurance Activity for FCS_CKM.2.1 in the GPOS PP for evaluating this SFR. Note that because a TOE that conforms to this PP-Module must implement IPsec, the tested protocols shall include IPsec at minimum.

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

[**selection**:

- ***AES-XTS (as defined in NIST SP 800-38E)***
- *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 SP 800-38C and IEEE 802.11ac-2013)*
- *AES-GCMP-256 (as defined in NIST SP 800-38D and IEEE 802.11ac-2013)*
- *No other modes*

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

Application Note: This SFR is defined in the GPOS PP as FCS_COP.1(1); the formatting of iteration convention was updated to be consistent with the PP-Module's conventions.

This SFR is identical to what is defined in the GPOS PP except that support for CBC and 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.

Evaluation Activities

FCS_COP.1/1
Refer to the EA for FCS_COP.1(1) in the GPOS PP for evaluating this SFR.

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 VPN Clients PP is claimed as the Base-PP.

5.1.2.1 Cryptographic Support (FCS)

FCS_CKM_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1

The [**selection, choose one of**: *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 or 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.

Evaluation Activities

FCS_CKM_EXT.2
TSS
Regardless of whether this requirement is met by the VPN client or the OS, the evaluator will check the TSS to ensure that it lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored.

The evaluator shall review the TSS for to determine that it makes a case that, for each item listed as being manipulated, it is not written unencrypted to persistent memory, and that the item is stored by the OS.

Guidance
There are no guidance EAs for this requirement.

Tests
There are no test EAs for this component.

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, choose one of**: *VPN client*, *OS*] shall [**selection, choose one of**: *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 certificate revocation list (CRL) or to use the online certificate status protocol (OCSP) to check revocation status. 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, choose one of**: *VPN client*, *OS*] shall not establish an SA if a certificate or certificate path is deemed invalid.

Evaluation Activities

FIA_X509_EXT.3
The EAs below apply to FIA_X509_EXT.3.2. FIA_X509_EXT.3.1 is evaluated as part of FCS_IPSEC_EXT.1 (and conditionally as part of FPT_TUD_EXT.1 or FPT_TST_EXT.1) and FIA_X509_EXT.3.3 is evaluated as part of FCS_IPSEC_EXT.1.11.

TSS
The evaluator shall check the TSS to ensure that it describes whether the VPN client or the OS implements the certificate validation functionality, how the VPN client/OS chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the OS so that desired certificates can be used.

The evaluator shall examine the TSS to confirm that it describes the behavior of the client/OS when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Guidance
If the requirement indicates that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Tests
The evaluator shall perform the following test regardless of whether the certificate validation functionality is implemented by the VPN client or by the OS:

- *Test 1: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.3.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.*

5.1.2.3 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

The **[selection, choose one of: *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, choose one of: *VPN client*, *OS*]** shall permit [*the TSF*] to initiate communication with the trusted channel.

FTP_ITC.1.3

The **[selection, choose one of: *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 mode, tunnel mode, or both.

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.

Evaluation Activities

FTP_ITC.1
TSS
The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to a VPN gateway, VPN client, or IPsec-capable network device in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specification. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

Guidance
The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to a VPN gateway, VPN client, or IPsec-capable network device, and that it contains recovery instructions should a connection be unintentionally broken.

Tests
The evaluator shall perform the following tests:

- *Test 2: The evaluator shall ensure that the TOE is able to initiate communications with a VPN gateway, VPN client, IPsec-capable network device using the protocols specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- *Test 3: The evaluator shall ensure, for each communication channel with an IPsec peer, the channel data is not sent in plaintext.*
- *Test 4: The evaluator shall ensure, for each communication channel with an IPsec peer, modification of the channel data is detected by the TOE.*
- *Test 5: The evaluator shall physically interrupt the connection from the TOE to the IPsec peer. The evaluators shall ensure that subsequent communications are appropriately protected, at a minimum in the case of any attempts to automatically resume the connection or connect to a new access point.*

Further EAs are associated with requirements for FCS_IPSEC_EXT.1.

5.2 VPN Clients PP Security Functional Requirements Direction

In a PP-Configuration that includes the MDF PP, the VPN client is expected to rely on some of the security functions implemented by the OS as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.5.

5.2.1 Modified SFRs

The SFRs listed in this section are defined in the VPN Clients PP 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" [selection: *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 [**selection**:*
  - *cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.1*
  - *Diffie-Hellman group 14 that meet the following: RFC 3526*
  - *"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

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3*
- ***ECC schemes using 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 MDF PP except that ECC key generation with support for at least one of P-256 and P-384 has been made mandatory in support of IPsec due to the mandated support for at least one of DH groups 19 and 20 in FCS_IPSEC_EXT.1.8. Support for "safe-prime" groups has also been added as a selectable option for DH groups that use finite field algorithms. Curve25519 schemes remain selectable for their potential use in satisfying FDP_DAR_EXT.2.2 in the MDF PP; these schemes are not used in support of IPsec. RSA and ECC support for P-521 remain present as selections since they may be used by parts of the TOE that are not specifically related to VPN client functionality.

Evaluation Activities

FCS_CKM.1
Refer to the EAs for FCS_CKM.1 in the MDF PP. The only change to this SFR is that some selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

FCS_CKM.2/UNLOCKED Cryptographic Key Establishment

FCS_CKM.2.1/UNLOCKED

The TSF shall 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 Revision 3, "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 Revision 3, "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*
- *RSA-based key establishment schemes that meet the following: [**selection**:*
  - *NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes using Integer Factorization Cryptography"*
  - *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*

]

- ***no other key establishment schemes***

].

Application Note: This SFR differs from its definition in the MDF 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). Finite field and Group 14 selections remain present if groups 14, 15, or 24 are selected in FCS_IPSEC_EXT.1.8. This PP-Module does not require the use of RSA for any function but it is present in the selection in case other MDF PP functions require its use.

Evaluation Activities

FCS_CKM.2/UNLOCKED
Refer to the EAs for FCS_CKM.2/UNLOCKED in the MDF PP. The only change to this SFR is that some selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

FCS_COP.1/ENCRYPT Cryptographic Operation

FCS_COP.1.1/ENCRYPT

The TSF shall perform encryption/decryption 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 SP 800-38C and IEEE 802.11ac-2013)*
- *AES-GCMP-256 (as defined in NIST SP 800-38D 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 identical to what is defined in the MDF 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.

Evaluation Activities

FCS_COP.1/ENCRYPT
Refer to the EAs for FCS_COP.1/ENCRYPT in the MDF PP. The only change to this SFR is that some selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

5.2.1.2 User Data Protection (FDP)

FDP_IFC_EXT.1 Subset Information Flow Control

FDP_IFC_EXT.1.1

The TSF shall [

- *provide a VPN client which can protect all IP traffic using IPsec as defined in the PP-Module for VPN Client*

] with the exception of IP traffic needed to manage the VPN connection, and [**selection**: *[**assignment**: traffic needed for correct functioning of the TOE], no other traffic*] when the VPN is enabled.

Application Note: This SFR is identical to its definition in the Base-PP except that the selection item that requires the TOE to implement its own VPN client is always selected when the TOE's conformance claim includes this PP-Module

Evaluation Activities

FDP_IFC_EXT.1
Refer to the EAs for FDP_IFC_EXT.1 in the MDF PP. The only change to this SFR is that some selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

5.2.1.3 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 mutually authenticated TLS as defined in the Package for Transport Layer Security, HTTPS, IPsec in accordance with the PP-Module for VPN Client, [**selection**: *mutually authenticated DTLS as defined in the Package for Transport Layer Security, 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 validity of a certificate, the TSF shall [**selection, choose one of**: *allow the administrator to choose whether to accept the certificate in these cases, allow the user 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 MDF PP except that support for IPsec is mandated. The selection of “no other protocols” is added to address the case where the TOE only claims support for the protocols that are mandated by the SFR.

Evaluation Activities

FIA_X509_EXT.2
Refer to the EAs for FIA_X509_EXT.2 in the MDF PP. The only change to this SFR is that some selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

5.2.1.4 Security Management (FMT)

FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1

This SFR is not reproduced in its entirety for size purposes. The only change to this SFR is the following change to management function 45:

45. enable/disable the Always On VPN protection: - a. across device - [*d. no other method*]	**M**	O	O	O

Evaluation Activities

FMT_SMF_EXT.1
Refer to the EAs for FMT_SMF_EXT.1 in the MDF PP. The only change to this SFR is the change to management function 45. Testing of all other functions is not affected.

5.2.1.5 Trusted Path/Channels (FTP)

FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1.1

The TSF shall use

- 802.11-2012 in accordance with the Extended Package for WLAN Clients
- 802.1X in accordance with the Extended Package for WLAN Clients
- EAP-TLS in accordance with the Extended Package for WLAN Clients
- mutually authenticated TLS as defined in the Package for Transport Layer Security
- **IPsec in accordance with the PP-Module for VPN Client**

and [**selection**: *mutually authenticated DTLS as defined in the Package for Transport Layer Security, HTTPS, **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 Base-PP except that support for IPsec is mandated. Additionally, since the Base-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.

Evaluation Activities

FTP_ITC_EXT.1
Refer to the EAs for FTP_ITC_EXT.1 in the MDF PP. The only change to this SFR is that some

selections are mandated, therefore the corresponding testing is mandatory. The actual testing for those selections is not changed.

5.2.2 Additional SFRs

This PP-Module does not define any additional SFRs for any PP-Configuration where the VPN Clients PP is claimed as the Base-PP.

5.3 App PP Security Functional Requirements Direction

In a PP-Configuration that includes the App PP, the VPN client is expected to rely on some of the security functions implemented by the OS as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.5.

5.3.1 Modified SFRs

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

5.3.1.1 Cryptographic Support (FCS)

FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK

The application shall [**selection, choose one of**:

- *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], and,**

[**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*
- *[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]*
- *[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 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 App PP except that ECC key generation has been made mandatory in support of IPsec due to the mandated support for DH groups 19, and 20 in FCS_IPSEC_EXT.1.8. RSA remains present as a selection since it may be used by parts of the TOE that are not specifically related to VPN client functionality.

Evaluation Activities

FCS_CKM.1/AK
Refer to the EA for FCS_CKM.1/AK in the App PP.

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

The application shall [**selection, choose one of**: *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*"]; and**

[**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]*
- *[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]*
- *[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 2.1*
- *[RSA-based key establishment schemes] that meet the following: [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"]*
- *No other schemes*

].

Application Note: This SFR differs from its definition in the App 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 at least one of NIST SP 800-56A, RFC 3526, or NIST SP 800-56A rev. 3 "safe-prime" groups for key establishment using finite field cryptography.

Evaluation Activities

FCS_CKM.2
For all key establishment schemes refer to the EA for FCS_CKM.2 in the App PP.

FCS_CKM.1 Cryptographic Key Generation Services

FCS_CKM.1.1

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

Application Note: This selection differs from its definition in the App PP 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/AK to be claimed.

Evaluation Activities

FCS_CKM.1
This SFR is evaluated in conjunction with FCS_CKM.1/AK in the App PP.

FCS_COP.1/SKC Cryptographic Operation

FCS_COP.1.1/SKC

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*
- *AES-CCM (as defined in NIST SP 800-38C) mode*
- *AES-CTR (as defined in NIST SP 800-38A) mode*
- ***no other modes***

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

Application Note: This SFR is selection-based in the Base-PP and remains selection-based here because this PP-Module allows for the possibility that the TSF relies on platform-provided cryptographic algorithm services for its own implementation of IPsec. However, if the TSF does claim this SFR to support IPsec, the ST author must select at minimum both AES-CBC and AES-GCM with both 128-bit and 256-bit key sizes for consistency with the relevant IPsec claims (FCS_IPSEC_EXT.1.4 requires both 128-bit and 256-bit AES-GCM and FCS_IPSEC_EXT.1.6 requires both 128-bit and 256-bit AES-CBC).

Evaluation Activities

FCS_COP.1/SKC
TSS
If the TSF implements AES cryptography in support of both credential encryption (per FCS_STO_EXT.1) and IPsec, the evaluator shall examine the TSS to ensure that it clearly identifies the modes and key sizes that are supported for each usage of AES.

Guidance
There are no operational beyond what is required by the EA for FCS_COP.1/SKC in the App PP.

Tests
There are no test EAs beyond what is required by the EA for FCS_COP.1/SKC in the App PP.

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 [**no other protocols**].

FIA_X509_EXT.2.2

When the application cannot establish a connection to determine the validity of a certificate, the TSF shall [**selection, choose one of**: *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 App PP 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 since it is expected that IPsec is the TOE's only use of certificates.

Evaluation Activities

FIA_X509_EXT.2
Refer to the EA for FIA_X509_EXT.2 in the App PP.

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] using IPsec as specified in FCS_IPSEC_EXT.1 and [**selection**: *HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client*, *HTTPS as a server in accordance with FCS_HTTPS_EXT.1/Server*, *HTTPS as a server with mutual authentication in accordance with FCS_HTTPS_EXT.2*, *TLS as defined in the Functional Package for TLS*, *DTLS as defined in the Functional Package for TLS*, *SSH as defined in the Functional Package for Secure Shell*, *no other protocols*] between itself and another trusted IT product.

Application Note: This SFR is identical to what is defined in the App PP except that mandatory support for IPsec is added, the ST author is forced to select the 'encrypt all transmitted sensitive data' option, and the options for invoking platform-provided functionality have been removed. Since it is possible that a conformant TOE may not use any encryption protocols other than IPsec, "no other protocols" is provided as a selectable option in the list of supported protocols.

Evaluation Activities

FTP_DIT_EXT.1
For IPsec, refer to the EA for FCS_IPSEC_EXT.1. If other protocols are selected for FTP_DIT_EXT.1, refer to the EA for FTP_DIT_EXT.1 in the App PP.

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_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1

The [**selection, choose one of**: *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 Base-PP, which only applies to secure storage of administrative credentials. If some secrets or 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.

Evaluation Activities

FCS_CKM_EXT.2
TSS
Regardless of whether this requirement is met by the TOE or the TOE platform, the evaluator will check the TSS to ensure that it lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored. The evaluator then performs the following actions:

Persistent secrets and private keys manipulated by the platform:

For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the persistent secrets and private keys listed as being stored by the platform in the VPN client ST are identified as being protected in that platform's ST

Persistent secrets and private keys manipulated by the TOE:

The evaluator reviews the TSS for to determine that it makes a case that, for each item listed as being manipulated by the TOE, it is not written unencrypted to persistent memory, and that the item is stored by the platform.

Guidance
There are no guidance EAs for this requirement.

Tests
There are no test EAs for this requirement.

FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1

The [**selection, choose one of**: *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 or CSP data (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key or 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 or 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 must select both options.

Evaluation Activities

FCS_CKM_EXT.4
TSS
The evaluator shall ensure that all plaintext secret and private cryptographic keys and CSPs (whether manipulated by the TOE or exclusively by the platform) are identified in the VPN Client ST's TSS, and that they are accounted for by the EAs in this section.

Requirement met by the platform:

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key that are not otherwise covered by the FCS_CKM_EXT.4 requirement levied on the TOE.

For each platform listed in the ST, the evaluator shall examine the TSS of the ST of the platform to ensure that each of the secret keys, private keys, and CSPs used to generate key listed above are covered.

Requirement met by the TOE:

The evaluator shall check to ensure the TSS describes when each of the plaintext keys are cleared (e.g., system power off, disconnection of an IPsec connection, when no longer needed by the VPN channel per the protocol); and the type of clearing procedure that is performed (cryptographic erase, overwrite with zeros, overwrite three or more times by a different alternating pattern, overwrite with random pattern, or block erase). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are cleared by overwriting once with zeros, while secret keys stored on the internal persistent storage device are cleared by overwriting three times with a random pattern that is changed before each write").

Guidance
There are no guidance EAs for this requirement.

Tests
For each key clearing situation described in the TSS, the evaluator shall repeat the following test.

- *Test 6: The evaluator shall use appropriate combinations of specialized OE and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.*

  Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate copies of keys that are persisted encrypted by the TOE:

  1. *Load the instrumented TOE build in a debugger.*
  2. *Record the value of the key in the TOE subject to clearing.*
  3. *Cause the TOE to perform a normal cryptographic processing with the key from #1.*
  4. *Cause the TOE to clear the key.*
  5. *Cause the TOE to stop the execution but not exit.*
  6. *Cause the TOE to dump the entire memory footprint of the TOE into a binary file.*
  7. *Search the content of the binary file created in #4 for instances of the known key value from #1.*

  The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise.

The evaluator shall perform this test on all keys, including those persisted in encrypted form, to ensure intermediate copies are cleared.

5.4 MDM PP Security Functional Requirements Direction

In a PP-Configuration that includes the MDM PP, the VPN client is expected to rely on some of the security functions implemented by the OS as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.5.

5.4.1 Modified SFRs

The SFRs listed in this section are defined in the MDM PP and relevant to the secure operation of the TOE.

5.4.1.1 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1

The TSF shall [**selection, choose one of**: *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 meets the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.4, and**

[**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.3*
- *FFC schemes using cryptographic key sizes of 2048-bit or greater that meets the following: FIPS PUB 186-4, "Digital Signature Standards (DSS)," Appendix B.4*
- *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]*
- ***FFC schemes using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3***
- ***No other key generation schemes***

].

Application Note: This SFR is modified from its definition in the MDM PP by mandating the key generation algorithms that are required by this PP-Module in support of IPsec due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8. Other selections may be chosen by the ST author as needed for parts of the TOE that are not specifically related to VPN client functionality.

Evaluation Activities

FCS_CKM.1
Refer to the EA for FCS_CKM.1 in the MDM PP.

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

The TSF shall [**selection, choose one of**: *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 Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**

and [**selection**:

- *RSA-based key establishment schemes that meet 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 2.1"*
- *Finite field-based key establishment schemes that meets the following:*

*NIST Special Publication 800-56A **Revision 3**, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm 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]*
- *Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3*
- ***No other schemes***

].

Application Note: This SFR is modified from its definition in the MDM PP by mandating the key establishment algorithms that are required by this PP-Module in support of IPsec due to the mandated support for DH groups 19 and 20 in FCS_IPSEC_EXT.1.8. Other selections may be chosen by the ST author as needed for parts of the TOE that are not specifically related to VPN client functionality.

Evaluation Activities

FCS_CKM.2
Refer to the EA for FCS_CKM.2 in the MDM PP.

FCS_COP.1/1 Cryptographic Operation

FCS_COP.1.1/1

The TSF shall [**selection, choose one of**: *invoke platform-provided functionality, implement functionality*] perform encryption/decryption in accordance with a specified cryptographic algorithm

- **AES-CBC (as defined in FIPS PUB 197, NIST SP 800-38A),**
- **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)*
- ***no other modes***

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

Application Note: This SFR is defined in the MDM PP as FCS_COP.1(1); the formatting of iteration convention was updated to be consistent with the PP-Module's conventions.

This SFR is modified from its definition in the Base-PP by mandating support for both 128-bit and 256-bit implementations of AES-CBC (which this PP-Module requires for the use of IKE and allows for the use of ESP) and AES-GCM (which this PP-Module requires for the use of ESP and allows for the use of IKE). Other AES modes may be selected by the ST author as needed to address functions not required by this PP-Module.

Evaluation Activities

FCS_COP.1/1
Refer to the EA for FCS_COP.1(1) in the MDM PP.

5.4.1.2 Identification and Authentication (FIA)

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

The TSF shall [**selection**:

- *Invoke platform-provided functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [**selection**: ~~IPsec~~, HTTPS, TLS, DTLS, SSH, no protocols] and [**selection**:*
  - *code signing for system software updates*
  - *code signing for integrity verification*
  - *policy signing*
  - *[**assignment**: other uses]*
  - *no additional uses*

]

- *use X.509v3 certificates as defined by RFC 5280 to support authentication for*
  - ***IPsec as specified in the PP-Module for VPN client and***

  *[**selection**:*
  - *HTTPS in accordance with FCS_HTTPS_EXT.1*
  - *TLS as defined in the Package for Transport Layer Security*
  - *DTLS as defined in the Package for Transport Layer Security*
  - *SSH as defined in the Extended Package for Secure Shell*
  - *no other protocols*

  *], and [**selection**:*
  - *code signing for system software updates*
  - *code signing for integrity verification*
  - *policy signing*
  - *[**assignment**: other uses]*
  - *no additional uses*

  ]

].

Application Note: The PP-Module requires the TOE to implement its own X.509 authentication mechanism in support of IPsec communications. Other selections may be chosen by the ST author as needed for parts of the TOE that are not specifically related to VPN client functionality. The TSF may also rely on a platform-provided mechanism for uses of X.509 that do not relate to the establishment of trusted communications, as specified in the original SFR. FIA_X509_EXT.2.2 has not been included here as the PP-Module does not modify this element.

Evaluation Activities

FIA_X509_EXT.2
Refer to the EA for FIA_X509_EXT.2 in the MDM PP.

5.4.1.3 Protection of the TSF (FPT)

FPT_ITT.1/1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1/1

The TSF shall [*implement functionality using [IPsec as defined in the PP-Module for VPN Client]*].

Application Note: This SFR is defined in the MDM PP as FPT_ITT.1(1); the formatting of iteration convention was updated to be consistent with the PP-Module's conventions.

When the MDM TOE claims this PP-Module, at least one of its interfaces will implement IPsec communications. However, this PP-Module does not specify that any one particular interface must be implemented using IPsec. If the TOE is distributed and uses IPsec to secure communications between its distributed components, FPT_ITT.1(1) is refined as above.

This SFR is selection-based in the Base-PP depending on the selections made in the Base-PP requirement FTP_ITC_EXT.1. This is not changed by the PP-Module.

This SFR is modified from its definition in the Base-PP by mandating that the TSF implement IPsec communications and by prohibiting the TOE from relying on platform-provided functionality to implement this.

Evaluation Activities

FPT_ITT.1/1
Refer to the EA for FPT_ITT.1(1) in the MDM PP. Note that the PP-Module does not require any separate testing for this if IPsec is not used to implement this function.

5.4.1.4 Trusted Path/Channels (FTP)

FTP_ITC.1/1 Inter-TSF Trusted Channel (Authorized IT Entities)

FTP_ITC.1.1/1

The TSF shall **implement functionality using IPsec as defined in the PP-Module for VPN Client, and** [**selection**:

- *SSH as defined in the Extended Package for Secure Shell*
- *mutually authenticated TLS as defined in the Package for Transport Layer Security*
- *mutually authenticated DTLS as defined in the Package for Transport Layer Security*
- *HTTPS in accordance with FCS_HTTPS_EXT.1*
- ***no other protocols***

] and [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *SSH*
  - *mutually authenticated TLS*
  - *mutually authenticated DTLS*
  - *HTTPS*

  ]
- ***not invoke any platform-provided functionality***

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [**selection**: *authentication server, [**assignment**: other capabilities]*] that is logically distinct from other communication channels and provides assured identification of its end points and protection of channel data from modification and disclosure.

FTP_ITC.1.2/1

The TSF shall **implement functionality and** [**selection, choose one of**: *invoke platform-provided functionality*, ***not invoke platform-provided functionality***] to permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3/1

The TSF shall **implement functionality and** [**selection, choose one of**: *invoke platform-provided functionality*, ***not invoke platform-provided functionality***] to initiate communication via the trusted channel for [**assignment**: *list of services for which the TSF is able to initiate communications*].

Application Note: This SFR is defined in the MDM PP as FTP_ITC.1(1); the formatting of iteration convention was updated to be consistent with the PP-Module's conventions.

When the MDM TOE claims this PP-Module, at least one of its interfaces will implement IPsec communications. However, this PP-Module does not specify that any one particular interface must be implemented using IPsec. If the TOE uses IPsec to secure communications between itself and external trusted IT entities, FTP_ITC.1(1) is refined as noted by the refinements above.

This SFR is refined from its definition in the Base-PP by mandating that the “implement functionality” selection be chosen at minimum for IPsec and by prohibiting the TOE from relying on platform-provided IPsec functionality. Since the TOE may support multiple trusted channel interfaces, the ST author is given the option to select other protocols (SSH, TLS, DTLS, HTTPS) either as being implemented by the TSF or invoked from the platform.

Evaluation Activities

FTP_ITC.1/1
Refer to the EA for FTP_ITC.1(1) in the MDM PP. Note that the PP-Module does not require any separate testing for this if IPsec is not used to implement this function.

FTP_TRP.1/1 Trusted Path (for Remote Administration)

FTP_TRP.1.1/1

The TSF shall **implement functionality using IPsec as defined in the PP-Module for VPN Client, and [selection:**

- ***TLS as defined in the Package for Transport Layer Security***
- ***HTTPS in accordance with FCS_HTTPS_EXT.1***
- ***SSH as defined in the Extended Package for Secure Shell***
- ***no other protocols***

] and [**selection**:

- *invoke platform-provided functionality to use [**selection**:*

- *TLS*
- *HTTPS*
- *SSH*

]

- ***not invoke any platform-provided functionality***

] to provide a trusted communication channel between itself as a [**selection**: *server, peer*] and remote administrators that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from [*modification, disclosure*].

FTP_TRP.1.2/1

The TSF shall **implement functionality and** [**selection, choose one of**: *invoke platform-provided functionality,* ***not invoke platform-provided functionality***] to permit remote administrators to initiate communication via the trusted channel.

FTP_TRP.1.3/1

The TSF shall **implement functionality and** [**selection, choose one of**: *invoke platform-provided functionality,* ***not invoke platform-provided functionality***] to require the use of the trusted path for [*all remote administration actions*].

Application Note: This SFR is defined in the MDM PP as FTP_TRP.1(1); the formatting of iteration convention was updated to be consistent with the PP-Module's conventions.

When the MDM TOE claims this PP-Module, at least one of its interfaces will implement IPsec communications. However, this PP-Module does not specify that any one particular interface must be implemented using IPsec. If the TOE uses IPsec to secure communications between itself and trusted remote administrators, FPT_TRP.1(1) is refined as below.

This SFR is refined from its definition in the Base-PP by mandating that the “implement functionality” selection be chosen at minimum for IPsec and by prohibiting the TOE from relying on platform-provided IPsec functionality. Since the TOE may support multiple remote administrative interfaces, the ST author is given the option to select other protocols (SSH, TLS, HTTPS) either as being implemented by the TSF or invoked from the platform.

Evaluation Activities

FTP_TRP.1/1
Refer to the EA for FTP_TRP.1(1) in the MDM PP. Note that the PP-Module does not require any separate testing for this if IPsec is not used to implement this function.

5.4.2 Additional SFRs

This PP-Module does not define any additional SFRs for any PP-Configuration where the MDM PP is claimed as the Base-PP.

5.5 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.

5.5.1 Auditable Events for Mandatory SFRs

Table 2: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.1/VPN	No events specified	N/A
FCS_IPSEC_EXT.1	Decisions to DISCARD or BYPASS network packets processed by the TOE.	• Presumed identity of source subject. • The entry in the SPD that applied to the decision.
	Failure to establish an IPsec SA.	• Identity of destination subject. • Reason for failure.
	Establishment/Termination of an IPsec SA.	• Identity of destination subject. • Transport layer protocol, if applicable.

		• Source subject service identifier, if applicable. • Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	No events specified	N/A
FMT_SMF.1/VPN	Success or failure of management function.	No additional information.
FPT_TST_EXT.1/VPN	No events specified	N/A

5.5.2 Cryptographic Support (FCS)

FCS_CKM.1/VPN VPN Cryptographic Key Generation (IKE)

FCS_CKM.1.1/VPN

The TSF shall **[selection, choose one of: *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*] ~~that meet the following: [**assignment:** *list of standards*~~].

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 OE.

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.

Evaluation Activities

FCS_CKM.1/VPN
TSS
The evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked.

Guidance
There are no guidance EAs for this requirement.

Tests
If this functionality is implemented by the TSF, refer to the following EAs, depending on the TOE's claimed Base-PP:

- *GPOS PP: FCS_CKM.1*
- *MDF PP: FCS_CKM.1*
- *App PP: FCS_CKM.1/AK*
- *MDM PP: FCS_CKM.1*

FCS_IPSEC_EXT.1 IPsec

FCS_IPSEC_EXT.1.1

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

Application Note: 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. However, this is only the case when the platform is within the TOE boundary, as is the case where this PP-Module is being claimed on top of a general-purpose OS or a mobile device.

When the TOE is a standalone software application, the IPsec functionality must be implemented by the TSF, though it is permissible for the TSF to invoke cryptographic algorithm services from the TOE platform to support the TOE's implementation of IPsec. The TOE may also rely on the TOE platform for X.509 certificate validation services, though it is the responsibility of the TSF to take the proper action based on the validation response that is returned.

It is also permissible for the TSF to rely on low-level capabilities of the platform to perform enforcement and routing functions as a result of the policies the TSF maintains. For example, while the TSF must provide the capability to implement the Security Policy Database (SPD) abstraction, it is allowed for the TSF to depend on the platform-provided network stack to perform the low-level packet filtering and routing actions once the TSF has set up those rules as defined by the SPD.

While enforcement of the IPsec requirements must be implemented by the TSF, it is permissible for the TSF to receive configuration of the IPsec behavior from an environmental source, most notably a VPN gateway.

RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of an 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 is not required to provide an administrative interface to create or maintain an SPD.

As an alternative, a client may provide an interface that can be used by another application or network entity, such as a VPN gateway, as a means to establish and populate the SPD. In either of these cases (the client provides an administrative interface, or an API), while the client is expected to maintain the SPD abstraction, it is permitted for the low-level enforcement and routing activities to be implemented by platform capabilities (e.g., a network driver) as configured by the client.

FCS_IPSEC_EXT.1.2

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

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [*AES-GCM-128, 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 TSF 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 8784, RFC 8247, and [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions]*

].

FCS_IPSEC_EXT.1.6

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

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

FCS_IPSEC_EXT.1.7

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

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

]. 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 security association (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 hard-coded limits, but in this case it is required that an administrator be able to configure the values. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE. It is appropriate to refine the requirement in terms of number of MB or KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

FCS_IPSEC_EXT.1.8

The TSF shall ensure that IKE protocols implement DH Groups

- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and**

[**selection**:

- *[**selection**: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526*
- *[**selection**: 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS, **no other DH Groups**] according to RFC 5114*

].

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

FCS_IPSEC_EXT.1.9

The TSF shall generate the secret value x used in the IKE DH 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) numbers of bits that is at least twice the “bits of security” value associated with the negotiated DH group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General*] bits.

FCS_IPSEC_EXT.1.10

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

FCS_IPSEC_EXT.1.11

The TSF shall ensure that all IKE protocols perform peer authentication using [**selection**: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: *Pre-shared keys*, *Pre-shared Keys transmitted via EAP-TLS*, *Pre-shared Keys transmitted via EAP-TTLS with mutual authentication*, *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).

If any selection with “pre-shared keys” is selected, the selection-based requirement FIA_PSK_EXT.1 must be claimed.

When pre-shared keys are supported for IKEv2, at least one of ‘Pre-shared Keys transmitted via EAP-TLS’ or ‘Pre-shared Keys transmitted via EAP-TTLS’ is selected to indicate client verification using certificates in a mutually authenticated TLS handshake, and verification of provided PSK protected under the TLS channel. The selection-based SFR FCS_EAP_EXT.1 must also be claimed in this situation.

When Pre-shared Keys are supported for IKEv1, the first selection is claimed to indicate one of the mechanisms for using PSK described in the RFC. It is acceptable for different use cases to leverage different selections, if this is the case it must be identified.

FCS_IPSEC_EXT.1.12

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

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

second selection.

FCS_IPSEC_EXT.1.13

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

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

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

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

FCS_IPSEC_EXT.1.14

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

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

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

Evaluation Activities

FCS_IPSEC_EXT.1
TSS
In addition to the TSS EAs for the individual FCS_IPSEC_EXT.1 elements below, the evaluator shall perform the following:

If the TOE boundary includes a general-purpose OS or mobile device, the evaluator shall examine the TSS to ensure that it describes whether the VPN client capability is architecturally integrated with the platform itself or whether it is a separate executable that is bundled with the platform.

Guidance
In addition to the Operational for the individual FCS_IPSEC_EXT.1 elements below, the evaluator shall perform the following:

If the configuration of the IPsec behavior is from an environmental source, most notably a VPN gateway (e.g through receipt of required connection parameters from a VPN gateway), the evaluator shall ensure that the operational guidance contains any appropriate information for ensuring that this configuration can be properly applied.

Note in this case that the implementation of the IPsec protocol must be enforced entirely within the TOE boundary; i.e. it is not permissible for a software application TOE to be a graphical front-end for IPsec functionality implemented totally or in part by the underlying OS platform.

The behavior referenced here is for the possibility that the configuration of the IPsec connection is initiated from outside the TOE, which is permissible so long as the TSF is solely responsible for enforcing the configured behavior. However, it is allowable for the TSF to rely on low-level platform-provided networking functions to implement the SPD from the client (e.g., enforcement of packet routing decisions).

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

The evaluator shall minimally create a test environment equivalent to the test environment illustrated below. 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 shall provide justification for any differences in the test environment.

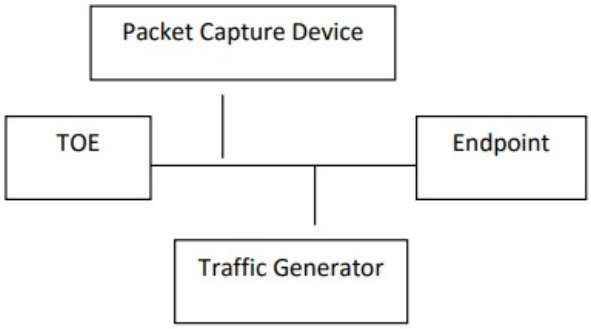


Figure 4: Test Environment

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

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

If the TOE is a standalone software application, the evaluator shall ensure that the TSS asserts that all IPsec functionality is implemented by the TSF. The evaluator shall also ensure that the TSS identifies what platform functionality the TSF relies upon to support its IPsec implementation, if any (e.g. does it invoke cryptographic primitive functions from the platform's cryptographic library, enforcement of packet routing decisions by low-level network drivers).

If the TOE is part of a general-purpose desktop or mobile OS, the evaluator shall ensure that the TSS describes at a high level the architectural relationship between the VPN client portion of the TOE and the rest of the TOE (e.g. is the VPN client an integrated part of the OS or is it a standalone executable that is bundled into the OS package). If the SPD is implemented by the underlying platform in this case, then the TSS describes how the client interacts with the platform to establish and populate the SPD, including the identification of the platform's interfaces that are used by the client.

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

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

Guidance
The evaluator shall examine the operational guidance to verify it describes how the SPD is

created and configured. If there is an administrative interface to the client, then the guidance describes how the administrator specifies rules for processing a packet. The description includes all three cases - a rule that ensures packets are encrypted/decrypted, dropped, and allowing a packet to flow in plaintext. The evaluator shall determine that the description in the operational guidance is consistent with the description in the TSS, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

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

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

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

The evaluator shall perform the following tests:

- *Test 7: The evaluator shall configure an SPD on the client that is capable of the following: dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the client with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed through without modification, was encrypted by the IPsec implementation.*
- *Test 8: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and operational guidance. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the operational guidance.*

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

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

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

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

- *Test 9: [conditional]: If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures a VPN gateway to operate in tunnel mode. The evaluator configures the TOE and the VPN gateway to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the client to connect to the VPN GW peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.*
- *Test 10: [conditional]: If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures an IPsec peer to accept IPsec connections using transport mode. The evaluator configures the TOE and the endpoint device to use any of the allowed cryptographic algorithms, authentication*

methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the remote endpoint. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.

- *Test 11: [conditional]: If both tunnel mode and transport mode are selected, the evaluator shall perform both Test 1 and Test 2 above, demonstrating that the TOE can be configured to support both modes.*
- *Test 12: [conditional]: If both tunnel mode and transport mode are selected, the evaluator shall modify the testing for FCS_IPSEC_EXT.1 to include the supported mode for SPD PROTECT entries to show that they only apply to traffic that is transmitted or received using the indicated mode.*

FCS_IPSEC_EXT.1.3

TSS

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

Guidance

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

Tests

The evaluator shall perform the following test:

- *Test 13: The evaluator shall configure the SPD such that it has entries that contain operations that DISCARD, PROTECT, and (if applicable) BYPASS network packets. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE-created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE's interfaces.*

FCS_IPSEC_EXT.1.4

TSS

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

Guidance

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

Tests

- *Test 14: The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the AES-GCM-128, and AES-GCM-256 algorithms, and attempt to establish a connection using ESP. If the ST Author has selected either AES-CBC-128 or AES-CBC-256, the TOE is configured to use those algorithms and the evaluator attempts to establish a connection using ESP for those algorithms selected.*

FCS_IPSEC_EXT.1.5

TSS

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

Guidance

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

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

Tests

- *Test 15: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 7296, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed. If the TOE supports IKEv1 with or without XAUTH, the evaluator shall verify that this test can be successfully repeated with XAUTH enabled and disabled in the manner specified by the operational guidance. If the TOE only supports IKEv1 with XAUTH, the evaluator shall verify that connections not using XAUTH are unsuccessful. If the TOE only supports IKEv1 without XAUTH, the evaluator shall verify that connections using XAUTH are unsuccessful.*
- *Test 16: [conditional]: If the TOE supports IKEv1, the evaluator shall perform any applicable operational guidance steps to disable the use of aggressive mode and then attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator shall show that the TOE will reject a VPN gateway from initiating an IKEv1 Phase 1 connection in aggressive mode. The evaluator should then show that main mode exchanges are supported.*

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

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

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

- *Test 17: The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKE payload for each supported IKE version and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation. The evaluator will confirm that the connection is successful by confirming that data can be passed through the connection once it is established. For example, the evaluator may connect to a webpage on the remote network and verify that it can be reached.*

FCS_IPSEC_EXT.1.7
TSS
There are no TSS EAs for this requirement.

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

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

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

- *Test 18: [conditional]: The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is*

exceeded, the connection is closed.

- *Test 19: [conditional]: The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.*
- *Test 20: [conditional]: The evaluator shall perform a test similar to Test 2 for Phase 2 SAs, except that the lifetime will be 8 hours or less instead of 24 hours or less.*
- *Test 21: [conditional]: If a fixed limit for IKEv1 SAs is supported, the evaluator shall establish an SA and observe that the connection is closed after the fixed traffic or time value is reached.*

FCS_IPSEC_EXT.1.8
TSS
The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Guidance
There are no guidance EAs for this requirement.

Tests
The evaluator shall perform the following test:

- *Test 22: For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.*

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

Guidance
There are no guidance EAs for this requirement.

Tests
There are no test EAs for this requirement.

FCS_IPSEC_EXT.1.10
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.9.
FCS_IPSEC_EXT.1.11
TSS
The evaluator shall ensures that the TSS whether peer authentication is performed using RSA, ECDSA, or both.

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

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

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

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

The evaluator ensures the operational guidance describes how to set up the TOE to use the cryptographic algorithms RSA, ECDSA, or either, depending which is claimed in the ST.

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

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

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

- *Test 23: The evaluator shall have the TOE generate a public-private key pair, and submit a CSR (Certificate Signing Request) to a CA (trusted by both the TOE and the peer VPN used to establish a connection) for its signature. The values for the DN (Common Name, Organization, Organizational Unit, and Country) will also be passed in the request. Alternatively, the evaluator may import to the TOE a previously generated private key and corresponding certificate.*
- *Test 24: The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.*
- *Test 25: The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. For this current version of the PP-Module, the evaluator has to only test one up in the trust chain (future drafts may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the SA is established. The evaluator then attempts the test with a certificate that will be revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the TOE will not establish an SA*
- *Test 26: [conditional]: For each selection made, the evaluator shall verify factors are required, as indicated in the operational guidance, to establish an IPsec connection with the server.*
  For each supported identifier type (excluding DNs), the evaluator shall repeat the following tests:
- *Test 27: For each field of the certificate supported for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds.*
- *Test 28: For each field of the certificate support for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to not match the field in the peer's presented certificate and shall verify that the IKE authentication fails.*
  The following tests are conditional:
- *Test 29: [conditional]: If, according to the TSS, the TOE supports both Common Name and SAN certificate fields and uses the preferred logic outlined in the Application Note, the tests above with the Common Name field shall be performed using peer certificates with no SAN extension. Additionally, the evaluator shall configure the peer's reference identifier on the TOE to not match the SAN in the peer's presented certificate but to match the Common Name in the peer's presented certificate, and verify that the IKE authentication fails.*
- *Test 30: [conditional]: If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds. To demonstrate a bit-wise comparison of the DN, the evaluator shall change a single bit in the DN (preferably, in an Object Identifier (OID) in the DN) and verify that the IKE authentication fails.* ***To demonstrate a comparison of DN values, the evaluator shall change any one of the four DN values and verify that the IKE authentication fails.***
- *Test 31: [conditional]: If the TOE supports both IPv4 and IPv6 and supports IP address identifier types, the evaluator must repeat test 1 and 2 with both IPv4 address identifiers and IPv6 identifiers. Additionally, the evaluator shall verify that the TOE verifies that the IP header matches the identifiers by setting the presented identifiers and the reference identifier with the same IP address that differs from the actual IP address of the peer in the IP headers and verifying that the IKE authentication fails.*
- *Test 32: [conditional]: If, according to the TSS, the TOE performs comparisons between the peer's ID payload and the peer's certificate, the evaluator shall repeat the following test for each combination of supported identifier types and supported certificate fields (as above). The evaluator shall configure the peer to present a different ID payload than the field in the peer's presented certificate and verify that the TOE fails to authenticate the IKE peer.*

FCS_IPSEC_EXT.1.12
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.11.
FCS_IPSEC_EXT.1.13
EAs for this element are tested through EAs for FCS_IPSEC_EXT.1.11.
FCS_IPSEC_EXT.1.14

TSS
The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 or IKEv2 CHILD_SA suites (depending on the supported IKE versions) to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Guidance
There are no guidance EAs for this requirement.

Tests
The evaluator follows the guidance to configure the TOE to perform the following tests:

- *Test 33: This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.*
- *Test 34: [conditional]: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.*
- *Test 35: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.*
- *Test 36: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.*

5.5.3 User Data Protection (FDP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1

The **[selection, choose one of: *TOE, TOE platform*]** shall ensure 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.

Evaluation Activities

FDP_RIP.2
TSS
Requirement met by the platform

The evaluator shall examine the TSS to verify that it describes (for each supported platform) the extent to which the client processes network packets and addresses the FDP_RIP.2 requirement.

Requirement met by the TOE

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

Guidance
There are no guidance EAs for this requirement.

Tests
There are no test EAs for this requirement.

5.5.4 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 cases supported by the TOE should be claimed. "Client credentials" will include the client certificate used for IPsec authentication, and may also include a PSK.

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.

Evaluation Activities

FMT_SMF.1/VPN
TSS
The evaluator shall check to ensure the TSS describes the client credentials and how they are used by the TOE.

Guidance
The evaluator shall check to make sure that every management function mandated in the ST for this requirement is described in the operational guidance and that the description contains the information required to perform the management duties associated with each management function.

Tests
The evaluator shall test the TOE’s ability to provide the management functions by configuring the TOE according to the operational guidance and testing each management activity listed in the ST.

The evaluator shall ensure that all management functions claimed in the ST can be performed by completing activities described in the AGD. Note that this may be performed in the course of completing other testing.

5.5.5 Protection of the TSF (FPT)

FPT_TST_EXT.1/VPN TSF Self-Test

FPT_TST_EXT.1.1/VPN

The [**selection, choose one of**: *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/VPN

The [**selection, choose one of**: *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 OE*].

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

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.

Evaluation Activities

FPT_TST_EXT.1/VPN
Except for where it is explicitly noted, the evaluator is expected to check the following information regardless of whether the functionality is implemented by the TOE or by the TOE platform.

TSS
The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on startup; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested," a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. If some of the tests are performed by the TOE platform, the evaluator shall check the TSS to ensure that those tests are identified, and that the ST for each platform contains a description of those tests. Note that the tests that are required by this component are those that support security functionality in the VPN Client PP-Module, which may not correspond to the set of all self-tests contained in the platform STs.

The evaluator shall examine the TSS to ensure that it describes how the integrity of stored TSF executable code is cryptographically verified when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator shall check to ensure that the cryptographic requirements listed are consistent with the description of the integrity verification process.

The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases. For checks implemented entirely by the platform, the evaluator ensures that the operational guidance for the TOE references or includes the platform-specific guidance for each platform listed in the ST.

Guidance
If not present in the TSS, the evaluator ensures that the operational guidance describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases. For checks implemented entirely by the platform, the evaluator ensures that the operational guidance for the TOE references or includes the platform-specific guidance for each platform listed in the ST.

Tests
The evaluator shall perform the following tests:

- *Test 37: The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful.*
- *Test 38: The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.*

5.6 TOE Security Functional Requirements Rationale

The following rationale provides justification for each security objective for the TOE, showing that the SFRs are suitable to meet and achieve the security objectives:

Table 3: SFR Rationale

Objective	Addressed by	Rationale
O.AUTHENTICATION	FIA_X509_EXT.3 (when GPOS PP is Base-PP)	This SFR supports the objective by enforcing the use of X.509 certificate authentication for IPsec.
	FDP_IFC_EXT.1 (refined from MDF PP)	This SFR supports the objective by affirming that the TOE includes a VPN client.
	FIA_X509_EXT.2 (refined from MDF PP)	This SFR supports the objective by enforcing the use of X.509 certificate authentication for IPsec.
	FIA_X509_EXT.2 (refined from App PP)	This SFR supports the objective by enforcing the use of X.509 certificate authentication for IPsec.
	FIA_X509_EXT.2 (refined from MDM PP)	This SFR supports the objective by enforcing the use of X.509 certificate authentication for IPsec.
	FCS_IPSEC_EXT.1	This SFR supports the objective by requiring the TOE's implementation of IPsec to include requirements for how the remote VPN gateway or peer is authenticated.
	FIA_BMA_EXT.1 (optional)	This SFR supports the objective by optionally defining the TOE's support for a platform-based biometric mechanism to use as an authentication mechanism.
	FPF_MFA_EXT.1 (optional)	This SFR supports the objective by optionally enforcing a multifactor authentication requirement on an IPsec connection.
	FCS_EAP_EXT.1 (selection-based)	This SFR supports the objective by optionally implementing EAP-TLS or EAP-TTLS as a mechanism for authentication.
	FIA_HOTP_EXT.1 (selection-based)	This SFR supports the objective by optionally defining the implementation of HOTP as an authentication mechanism.
	FIA_PSK_EXT.1 (selection-based)	This SFR supports the objective by optionally requiring support for pre-shared keys as an alternate authentication method for IPsec.
	FIA_PSK_EXT.2 (selection-based)	This SFR supports the objective by optionally specifying whether the TOE generates its own pre-shared keys used for authentication or accept them from an external source.
	FIA_PSK_EXT.3 (selection-based)	This SFR supports the objective by optionally defining the composition and use of password-based pre-shared keys used for authentication.
	FIA_PSK_EXT.4 (selection-based)	This SFR supports the objective by optionally defining HOTP as an authentication mechanism.
	FIA_PSK_EXT.5 (selection-based)	This SFR supports the objective by optionally defining TOTP as an authentication mechanism.
	FIA_TOTP_EXT.1 (selection-based)	This SFR supports the objective by optionally defining the implementation of TOTP as an authentication mechanism.
O.CRYPTOGRAPHIC_FUNCTIONS	FCS_CKM.1 (refined from GPOS PP)	This SFR supports the objective by requiring that the TOE implement key generation using certain methods.
	FCS_CKM.2 (refined from GPOS PP)	This SFR supports the objective by requiring that the TOE implement key establishment using certain methods.
	FCS_COP.1/1 (refined from GPOS PP)	This SFR supports the objective by requiring that the TOE implement symmetric encryption and decryption using certain methods.

FTP_ITC.1 (when GPOS PP is Base-PP)	This SFR supports the objective by requiring the TOE to support the use of IPsec as a trusted channel.
FCS_CKM.1 (refined from MDF PP)	This SFR supports the objective by requiring that the TOE implement key generation using certain methods.
FCS_CKM.2/UNLOCKED (refined from MDF PP)	This SFR supports the objective by requiring that the TOE implement key establishment using certain methods.
FCS_COP.1/ENCRYPT (refined from MDF PP)	This SFR supports the objective by requiring that the TOE implement symmetric encryption and decryption using certain methods.
FTP_ITC_EXT.1 (refined from MDF PP)	This SFR supports the objective by requiring the TOE to support the use of IPsec as a trusted channel.
FCS_CKM.1/AK (refined from App PP)	This SFR supports the objective by requiring the TOE to implement key generation using certain methods or to support invoking this function from its OS platform.
FCS_CKM.2 (refined from App PP)	This SFR supports the objective by requiring the TOE to implement key establishment using certain methods or to support invoking this function from its OS platform.
FCS_CKM_EXT.1 (refined from App PP)	This SFR supports the objective by requiring the TOE to specify whether it implements its own cryptographic primitives or invokes platform cryptographic services for these functions.
FCS_COP.1/SKC (refined from App PP)	This SFR supports the objective by requiring that the TOE implement symmetric encryption and decryption using certain methods.
FTP_DIT_EXT.1 (refined from App PP)	This SFR supports the objective by requiring the TOE to support the use of IPsec as a trusted channel.
FCS_CKM.1 (refined from MDM PP)	This SFR supports the objective by requiring the TOE to implement key generation using certain methods or to support invoking this function from its OS platform.
FCS_CKM.2 (refined from MDM PP)	This SFR supports the objective by requiring the TOE to implement key establishment using certain methods or to support invoking this function from its OS platform.
FCS_COP.1/1 (refined from MDM PP)	This SFR supports the objective by requiring that the TOE implement symmetric encryption and decryption using certain methods or invoke platform functionality that provides this capability.
FPT_ITT.1/1 (if applicable, refined from MDM PP)	If the MDM TOE includes a claim of this PP-Module to support protection of communications between distributed TOE components, this SFR supports the objective by requiring the TOE to support the use of IPsec for that interface.
FTP_ITC.1/1 (if applicable, refined from MDM PP)	If the MDM TOE includes a claim of this PP-Module to support protection of communications between the TOE and one or more trusted external IT entities, this SFR supports the objective by requiring the TOE to support the use of IPsec for that interface.
FTP_TRP.1/1 (if applicable, refined from MDM PP)	If the MDM TOE includes a claim of this PP-Module to support protection of communications between remote administrators and the TOE, this SFR supports the objective by requiring the TOE to support the use of IPsec for that interface.
FCS_CKM.1/VPN	This SFR supports the objective by requiring the TOE to generate keys used for IKE using certain methods.
FCS_IPSEC_EXT.1	This SFR supports the objective by requiring the TOE to implement the IPsec protocol in the specified manner.
FCS_EAP_EXT.1 (selection-based)	This SFR supports the objective by optionally defining the TOE's implementation of EAP-TLS or EAP-TTLS.

O.KNOWN_STATE	FMT_SMF_EXT.1 (refined from MDF PP)	This SFR supports the objective by requiring the portion of the TOE described by the Base-PP to include a management capability for the VPN client.
	FMT_SMF.1/VPN	This SFR supports the objective by requiring the TOE to implement certain administratively-configurable functions.
	FPT_TST_EXT.1/VPN	This SFR supports the objective by requiring the TOE to execute self-tests that demonstrate that its integrity is maintained.
	FAU_GEN.1/VPN (optional)	This SFR supports the objective by optionally requiring the TOE to generate audit records of its behavior.
	FAU_SEL.1/VPN (optional)	This SFR supports the objective by optionally requiring the TOE to allow for the configuration of what behavior is audited.
O.NONDISCLOSURE	FCS_CKM_EXT.2 (when GPOS PP is Base-PP)	This SFR supports the objective by requiring the TOE to store sensitive data in the OS’ key storage.
	FCS_CKM_EXT.2 (when App PP is Base-PP)	This SFR supports the objective by requiring the TOE or its platform to store sensitive data in the OS' key storage.
	FCS_CKM_EXT.4 (when App PP is Base-PP)	This SFR supports the objective by requiring the TOE or its platform to zeroize key data when no longer needed.
	FDP_RIP.2	This SFR supports the objective by requiring the TOE or its platform to ensure that residual data is purged from the system.
	FDP_VPN_EXT.1 (optional)	This SFR supports the objective by optionally requiring the TOE to prohibit split-tunneling so that network traffic cannot be transmitted outside of an established IPsec tunnel.
	FPF_MFA_EXT.1 (optional)	This SFR supports the objective by optionally requiring the TOE to prohibit transmission of packet data aside from those packets needed to perform multifactor authentication.

6 Consistency Rationale

6.1 Protection Profile for VPN Clients

6.1.1 Consistency of TOE Type

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

6.1.2 Consistency of Security Problem Definition

The threats and assumptions defined by this PP-Module (see sections 3.1 and 3.2) supplement those defined in the GPOS PP as follows:

PP-Module Threat, Assumption, OSP	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 GPOS PP.
T.TSF_CONFIGURATION	The threat of a misconfigured VPN client is consistent with the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats on the GPOS PP because misconfiguration could allow VPN traffic to be subjected unexpectedly to unauthorized modification or disclosure..
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.NETWORK_EAVESDROP threat in the GPOS 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 GPOS PP.
A.NO_TOE_BYPASS	The A.NO_TOE_BYPASS assumption assumes that the OE is configured in such a manner that the only network route to the protected network is through the TOE. This does not conflict with the GPOS PP because the GPOS PP makes no assumptions about the network architecture in which the TOE is deployed.
A.PHYSICAL	The assumption that physical security is provided by the environment is not explicitly stated in the GPOS PP but is consistent with the A.PLATFORM assumption defined in the GPOS PP, which expects the computing platform to be trusted.
A.TRUSTED_CONFIG	The assumption that personnel responsible for the TOE’s configuration are trusted to follow the guidance is consistent with the A.PROPER_ADMIN defined in the GPOS PP.

6.1.3 Consistency of Objectives

The security objectives defined by this PP-Module (see sections 4.1 and 4.2) supplement those defined in the GPOS PP as follows: The objectives for the TOEs are consistent with the VPN Clients PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.AUTHENTICATION	This objective is consistent with the O.PROTECTED_COMMS objective of the GPOS PP, which also expects that trusted remote channels will enforce authentication of remote endpoints.
O.CRYPTOGRAPHIC_FUNCTIONS	This objective is consistent with the O.PROTECTED_COMMS objective of the GPOS PP, which also expects that secure cryptographic functions are used to implement trusted communications.
O.KNOWN_STATE	This objective is consistent with the O.INTEGRITY objective of the GPOS PP, which expects a conformant TOE to implement measures to maintain its own integrity.
O.NONDISCLOSURE	This objective is consistent with the O.PROTECTED_STORAGE objective of the GPOS PP, which ensures that sensitive data is not disclosed without authorization.

The objectives for the TOE's OE are consistent with the VPN Clients PP based on the following rationale:

PP-Module OE

Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of the GPOS PP and does not define an environment that a GPOS TOE is incapable of existing in.
OE.PHYSICAL	This is part of satisfying OE.PLATFORM as defined in the GPOS 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 GPOS PP.

6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the VPN Clients PP that are needed to support VPN Client functionality. This is considered to be consistent because the functionality provided by the VPN Clients PP is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the VPN Clients PP as well as new SFRs that are used entirely to provide functionality for VPN Clients. The rationale for why this does not conflict with the claims defined by the VPN Clients 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 GPOS to satisfy any of its other security requirements.
Additional SFRs	
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 GPOS PP.
FIA_X509_EXT.3	This SFR defines additional uses for X.509 certificate functionality that do not conflict with those defined in the GPOS PP.
FTP_ITC.1	This SFR defines a trusted channel for IPsec, which is added functionality that does not prevent the existing GPOS functions from being performed.
Mandatory SFRs	
FCS_CKM.1/VPN	Generation of IKE peer authentication keys is added functionality that does not prevent the existing GPOS functions from being performed.
FCS_IPSEC_EXT.1	This SFR defines the VPN client’s IPsec implementation, which is added functionality that does not interfere with the GPOS 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 GPOS functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the GPOS as a whole can perform its security functions.
FPT_TST_EXT.1/VPN	Self-testing of the VPN client functionality does not impact the ability of the GPOS to perform its security functions.
Optional SFRs	
FIA_BMA_EXT.1	This SFR relates to biometric authentication, which does not conflict with the GPOS PP because it may be a function offered by the part of the TOE described by the GPOS PP.
FPF_MFA_EXT.1	This SFR relates specifically to the handling of traffic that is used for the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1/VPN	Audit records generated by the VPN client do not interfere with GPOS functionality. The possibility of the underlying OS platform generating audit records is consistent with the GPOS PP, which already contains FAU_GEN.1.

FAU_SEL.1/VPN	The ability to suppress the generation of certain audit records related to VPN activity does not interfere with the ability of the GPOS to satisfy its security functionality.
Implementation-based SFRs	
FDP_VPN_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 GPOS PP that would prevent this functionality from being supported.
Selection-based SFRs	
FCS_EAP_EXT.1	This SFR defines an additional cryptographic protocol that is beyond the scope of those defined in the GPOS PP but does not prevent any GPOS PP functionality from being implemented.
FIA_HOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
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.
FIA_PSK_EXT.2	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.3	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.4	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.5	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_TOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.

6.2 Protection Profile for VPN Clients

6.2.1 Consistency of TOE Type

If this PP-Module is used to extend the MDF 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 and assumptions defined by this PP-Module (see sections 3.1 and 3.2) supplement those defined in the MDF PP as follows:

PP-Module Threat, Assumption, OSP	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 MDF PP.
T.TSF_CONFIGURATION	The threat of a misconfigured VPN client is consistent with the T.NETWORK and T.EAVESDROP threats in the MDF PP because failure to mitigate against misconfiguration makes these threats more significant.
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.EAVESDROP threat in the MDF PP.
T.TSF_FAILURE	A failure of TSF functionality could compromise the local system, which is consistent with the T.FLAWAPP threat in the MDF PP.
A.NO_TOE_BYPASS	The A.NO_TOE_BYPASS assumption assumes that the OE is configured in such a manner that the only network route to the protected network is through the TOE. This does not conflict with the MDF PP because the MDF PP makes no assumptions about the network architecture in which the TOE is deployed.
A.PHYSICAL	The MDF PP includes the A.NOTIFY and A.PRECAUTION assumptions to mitigate the risk of physical theft of the TOE. This is consistent with the A.PHYSICAL assumption in this PP-Module because the MDF PP includes reasonable assumptions about the physical security of the TOE.

A.TRUSTED_CONFIG	This assumption is consistent with the MDF PP because the MDF PP includes the A.CONFIG assumption which assumes that all security functions are appropriately configured.

6.2.3 Consistency of Objectives

The security objectives defined by this PP-Module (see sections 4.1 and 4.2) supplement those defined in the MDF PP as follows: The objectives for the TOEs are consistent with the VPN Clients PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.AUTHENTICATION	This objective is consistent with the O.AUTH objective of the MDF PP, which also expects that trusted remote channels will enforce authentication of remote endpoints.
O.CRYPTOGRAPHIC_FUNCTIONS	This objective is consistent with the O.COMMS objective of the MDF PP, which also expects that secure cryptographic functions are used to implement trusted communications.
O.KNOWN_STATE	This objective is consistent with the O.INTEGRITY objective of the MDF PP, which expects a conformant TOE to implement measures to maintain its own integrity.
O.NONDISCLOSURE	This objective is consistent with the O.STORAGE objective of the MDF PP, which ensures that sensitive data is not disclosed without authorization.

The objectives for the TOE's OE are consistent with the VPN Clients PP based on the following rationale:

PP-Module OE Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of the MDF PP and does not define an environment that an MDF 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 MDF 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 MDF PP.

6.2.4 Consistency of Requirements

This PP-Module identifies several SFRs from the VPN Clients PP that are needed to support VPN Client functionality. This is considered to be consistent because the functionality provided by the VPN Clients PP is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the VPN Clients PP that are used entirely to provide functionality for VPN Clients. The rationale for why this does not conflict with the claims defined by the VPN Clients 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/UNLOCKED	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/ENCRYPT	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FDP_IFC_EXT.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.
FMT_SMF_EXT.1	This PP-Module modifies management function 45 regarding Always-on VPN protection.
FTP_ITC_EXT.1	This PP-Module adds IPsec as a new protocol that is used to implement trusted channels.

Additional SFRs	
This PP-Module does not add any requirements when the VPN Clients PP is the base.	
Mandatory 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 MDF PP.
FCS_IPSEC_EXT.1	This SFR defines the VPN client's IPsec implementation, which is added functionality that does not interfere with the MDF 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 MDF functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the MDF as a whole can perform its security functions.
FPT_TST_EXT.1/VPN	Self-testing of the VPN client functionality does not impact the ability of the MDF to perform its security functions
Optional SFRs	
FIA_BMA_EXT.1	This SFR relates to biometric authentication, which does not conflict with the MDF PP because it may be a function offered by the part of the TOE described by the MDF PP.
FPF_MFA_EXT.1	This SFR relates specifically to the handling of traffic that is used for the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1/VPN	Audit records generated by the VPN client do not interfere with MDF functionality. The possibility of the underlying MDF platform generating audit records is consistent with the MDF PP, which already contains FAU_GEN.1.
FAU_SEL.1/VPN	The ability to suppress the generation of certain VPN client audit records does not interfere with MDM functionality. The MDF PP already contains FAU_SEL.1 as an objective SFR which means that this functionality does not conflict with the expected behavior of a mobile device.
Implementation-based SFRs	
FDP_VPN_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 MDF PP that would prevent this functionality from being supported.
Selection-based SFRs	
FCS_EAP_EXT.1	This SFR defines an additional cryptographic protocol that is beyond the scope of those defined in the MDF PP but does not prevent any MDF PP functionality from being implemented.
FIA_HOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
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.
FIA_PSK_EXT.2	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.3	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.4	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.5	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_TOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.

6.3 Protection Profile for VPN Clients

6.3.1 Consistency of TOE Type

If this PP-Module is used to extend the App PP, 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 and assumptions defined by this PP-Module (see sections 3.1 and 3.2) supplement those defined in the App PP as follows:

PP-Module Threat, Assumption, OSP	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 misconfigured VPN client is consistent with the T.LOCAL_ATTACK threat in the App PP.
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.
A.NO_TOE_BYPASS	The A.NO_TOE_BYPASS assumption assumes that the OE is configured in such a manner that the only network route to the protected network is through the TOE. This does not conflict with the App PP because the App PP makes no assumptions about the network architecture in which the TOE is deployed.
A.PHYSICAL	The assumption that physical security is provided by the environment is not explicitly stated in the App PP but is consistent with the A.PLATFORM assumption defined in the App PP, which expects the computing platform to be trusted.
A.TRUSTED_CONFIG	The assumption that personnel responsible for the TOE’s configuration are trusted to follow the guidance is consistent with the A.PROPER_ADMIN defined in the App PP.

6.3.3 Consistency of Objectives

The security objectives defined by this PP-Module (see sections 4.1 and 4.2) supplement those defined in the App PP as follows: The objectives for the TOEs are consistent with the App PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.AUTHENTICATION	This objective is consistent with the O.PROTECTED_COMMS objective of the App PP, which also expects that trusted remote channels will enforce authentication of remote endpoints.
O.CRYPTOGRAPHIC_FUNCTIONS	This objective is consistent with the O.PROTECTED_COMMS objective of the App PP, which also expects that secure cryptographic functions are used to implement trusted communications.
O.KNOWN_STATE	This objective is consistent with the O.INTEGRITY objective of the App PP, which expects a conformant TOE to implement measures to maintain its own integrity.
O.NONDISCLOSURE	This objective is consistent with the O.PROTECTED_STORAGE objective of the App PP, which ensures that sensitive data is not disclosed without authorization.

The objectives for the TOE's OE are consistent with the App PP based on the following rationale:

PP-Module OE Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of the App PP 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 App PP 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 VPN Client functionality. This is considered to be consistent because the functionality provided by the App PP 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 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/AK	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.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/SKC	The ST author is given guidance to make specific selections if this selection-based SFR is claimed in support of IPsec functionality. 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 is for the VPN Client application and does not maintain any sensitive data of its own. Therefore, there is no need to protect (through FTP_DIT_EXT.1.1) VPN-client-specific data.
Additional SFRs	
FCS_CKM_EXT.2	This PP-Module adds a requirement for key storage, which is new functionality when compared to the App 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 App PP but does not interfere with its existing security functions.
Mandatory 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 App PP.
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/VPN	Self-testing of the VPN client functionality does not impact the ability of the application to perform its security functions
Optional SFRs	
FIA_BMA_EXT.1	This SFR relates to biometric authentication, which does not conflict with the App PP because it may be a function offered by the OE in which a TOE defined by the App PP is deployed.
FPF_MFA_EXT.1	This SFR relates specifically to the handling of traffic that is used for the establishment of IPsec connections.

Objective SFRs	
FAU_GEN.1/VPN	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 OS or mobile device, both of which can reasonably be expected to provide audit functionality.
FAU_SEL.1/VPN	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.
Implementation-based SFRs	
FDP_VPN_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.
Selection-based SFRs	
FCS_EAP_EXT.1	This SFR defines an additional cryptographic protocol that is beyond the scope of those defined in the App PP but does not prevent any App PP functionality from being implemented.
FIA_HOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
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.
FIA_PSK_EXT.2	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.3	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.4	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.5	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_TOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.

6.4 Protection Profile for VPN Clients

6.4.1 Consistency of TOE Type

If this PP-Module is used to extend the MDM PP, the TOE type for the overall TOE is still a mobile device management solution. The TOE boundary is simply extended to include VPN client functionality that is included with the MDM software so that additional security functionality is claimed within the scope of the TOE.

6.4.2 Consistency of Security Problem Definition

The threats and assumptions defined by this PP-Module (see sections 3.1 and 3.2) supplement those defined in the MDM PP as follows:

PP-Module Threat, Assumption, OSP	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 MDM PP.
T.TSF_CONFIGURATION	The threat of a misconfigured VPN client is consistent with the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats in the MDM PP because failure to mitigate against misconfiguration makes these threats more significant.
T.USER_DATA_REUSE	Inadvertent disclosure of user data to an unauthorized recipient is consistent with the T.NETWORK_EAVESDROP threat in the MDM PP
T.TSF_FAILURE	A failure of TSF functionality could compromise the implementation of the IPsec channel, which would lead to an exploitation of the

	T.NETWORK_ATTACK threat.
A.NO_TOE_BYPASS	The A.NO_TOE_BYPASS assumption assumes that the OE is configured in such a manner that the only network route to the protected network is through the TOE. This does not conflict with the MDM PP because the MDM PP makes no assumptions about the network architecture in which the TOE is deployed.
A.PHYSICAL	The assumption that physical security is provided by the environment is not explicitly stated in the MDM PP but is consistent with the A.MDM_SERVER_PLATFORM assumption defined in the MDM PP, which expects the computing platform to be trusted.
A.TRUSTED_CONFIG	The assumption that personnel responsible for the TOE's configuration are trusted to follow the guidance is consistent with the A.PROPER_ADMIN defined in the MDM PP.

6.4.3 Consistency of Objectives

The security objectives defined by this PP-Module (see sections 4.1 and 4.2) supplement those defined in the MDM PP as follows: The objectives for the TOEs are consistent with the MDM PP based on the following rationale:

PP-Module TOE Objective	Consistency Rationale
O.AUTHENTICATION	This objective is consistent with the O.DATA_PROTECTION_TRANSIT objective of the MDM PP, which also expects that trusted remote channels will enforce authentication of remote endpoints.
O.CRYPTOGRAPHIC_FUNCTIONS	This objective is consistent with the O.DATA_PROTECTION_TRANSIT objective of the MDM PP, which also expects that secure cryptographic functions are used to implement trusted communications.
O.KNOWN_STATE	This objective is consistent with the O.INTEGRITY objective of the MDM PP, which expects a conformant TOE to implement measures to maintain its own integrity.
O.NONDISCLOSURE	There are no objectives in the MDM PP that directly relate to this objective, but it could be considered to support both the O.ACCOUNTABILITY and O.MANAGEMENT objectives in the MDM PP by ensuring that stored data cannot be modified through unauthorized mechanisms that may allow for access control and logging functions to be bypassed.

The objectives for the TOE's OE are consistent with the MDM PP based on the following rationale:

PP-Module OE Objective	Consistency Rationale
OE.NO_TOE_BYPASS	This objective addresses behavior that is out of scope of the MDM PP and does not define an environment that an MDM TOE is incapable of existing in.
OE.PHYSICAL	This is part of satisfying OE.IT_ENTERPRISE as defined in the MDM PP because provisioning of physical security is a reasonable expectation for an IT enterprise.
OE.TRUSTED_CONFIG	The expectation of trusted configuration is consistent with OE.PROPER_USER and OE.PROPER_ADMIN in the MDM PP.

6.4.4 Consistency of Requirements

This PP-Module identifies several SFRs from the MDM PP that are needed to support VPN Client functionality. This is considered to be consistent because the functionality provided by the MDM PP is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the MDM PP that are used entirely to provide functionality for VPN Clients. The rationale for why this does not conflict with the claims defined by the MDM 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 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	The ST author is instructed to make specific selections at minimum to address VPN client requirements; the SFR behavior itself is unmodified.
FPT_ITT.1/1	When this SFR relates to the PP-Module's functionality, the ST author is instructed to make specific selections to implement this behavior using the VPN client. This is done by forcing the ST author to make specific selections that are already present in the MDM PP definition of the SFR; no new behavior is introduced by this.
FTP_ITC.1/1	When this SFR relates to the PP-Module's functionality, the ST author is instructed to make specific selections to implement this behavior using the VPN client at minimum. This is done by forcing the ST author to make a specific selection that is already present in the MDM PP definition of the SFR and by removing a selection option; no new behavior is introduced by this.
FTP_TRP.1/1	When this SFR relates to the PP-Module's functionality, the ST author is instructed to make specific selections to implement this behavior using the VPN client at minimum. This is done by forcing the ST author to make a specific selection that is already present in the MDM PP definition of the SFR and by removing a selection option; no new behavior is introduced by this.
Additional SFRs	
This PP-Module does not add any requirements when the MDM PP is the base.	
Mandatory 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 MDM PP
FCS_IPSEC_EXT.1	This SFR defines the VPN client's IPsec implementation, which is added functionality that does not interfere with the MDM 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 MDM functionality.
FMT_SMF.1/VPN	The ability to configure the VPN client behavior does not affect whether the MDM as a whole can perform its security functions.
FPT_TST_EXT.1/VPN	Self-testing of the VPN client functionality does not impact the ability of the MDM to perform its security functions.
Optional SFRs	
FIA_BMA_EXT.1	This SFR relates to biometric authentication, which does not conflict with the MDM PP because it may be a function offered by the part of the TOE described by the MDM PP.
FPF_MFA_EXT.1	This SFR relates specifically to the handling of traffic that is used for the establishment of IPsec connections.
Objective SFRs	
FAU_GEN.1/VPN	Audit records generated by the VPN client do not interfere with MDM functionality. The possibility of the MDM as a whole generating audit records is consistent with the MDM PP, which already contains FAU_GEN.1
FAU_SEL.1/VPN	The ability to suppress the generation of certain VPN client audit records does not interfere with MDM functionality. The MDM PP already contains FAU_SEL.1 as an optional SFR which means that this functionality does not conflict with the expected behavior of an MDM.
Implementation-based SFRs	
FDP_VPN_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 MDM PP that would prevent this functionality from being supported.
Selection-based SFRs	
FCS_EAP_EXT.1	This SFR defines an additional cryptographic protocol that is beyond the scope of those defined in the MDM PP but does not prevent any MDM PP functionality from being implemented.
FIA_HOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.

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.
FIA_PSK_EXT.2	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.3	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.4	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_PSK_EXT.5	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.
FIA_TOTP_EXT.1	This SFR relates to use of pre-shared keys, which is behavior that only applies to the establishment of IPsec connections.

Appendix A - Optional SFRs

A.1 Strictly Optional Requirements

A.1.1 Auditable Events for Strictly Optional SFRs

Table 4: Auditable Events for Strictly Optional Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FIA_BMA_EXT.1	No events specified	N/A
FPF_MFA_EXT.1	No events specified	N/A

A.1.2 Identification and Authentication (FIA)

The TOE may support leveraging the biometric API provided by the platform.

FIA_BMA_EXT.1 Biometric Activation

FIA_BMA_EXT.1.1

The TSF shall leverage the platform biometric features to confirm the user before initiating a trusted channel.

Application Note: In this context the platform refers to the OS or device and may be part of the TOE if those Base-PPs are leveraged.

Evaluation Activities

FIA_BMA_EXT.1
TSS
The evaluator shall confirm that the TSS describes the calls to the platform and verifies they align with platform documentation.

Guidance
The evaluator shall ensure that any configuration details needed to enable the biometric prompt are included in the guidance documentation.

Tests
- *Test 39: The evaluator shall initiate a connection and verify that a biometric prompt is presented and accepted before the connection can proceed. The evaluator shall also verify the connection does not proceed if the biometric is not presented or accepted.*

A.1.3 Packet Filtering (FPF)

FPF_MFA_EXT.1 Multifactor Authentication Filtering

FPF_MFA_EXT.1.1

The TSF shall not forward packets to the internal network until the IKE/IPsec tunnel has been established, except those necessary to ensure that the client is authenticated according to FIA_PSK_EXT.1.

Application Note: If FPF_MFA_EXT.1 is included FIA_PSK_EXT.1 must be included.

Evaluation Activities

FPF_MFA_EXT.1
TSS
The evaluator shall examine the TSS to verify that it describes how authentication packets are identified and how all other traffic is blocked until secondary authentication is successful.

Guidance
The evaluator shall examine the operational guidance to verify that it provides instructions to the administrator on how to configure the secondary HOTP or TOTP factors and any additional details necessary for filtering all other traffic.

Tests

- *Test 40: For each included selection the evaluator shall configure the TOE per the operational guidance. The evaluator shall attempt to connect and verify other traffic is rejected per the filtering rules. The evaluator shall then provide the selected factor and confirm it is accepted and traffic is no longer blocked.*

A.2 Objective Requirements

A.2.1 Auditable Events for Objective SFRs

Table 5: Auditable Events for Objective Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/VPN	No events specified	N/A
FAU_SEL.1/VPN	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional information.

A.2.2 Security Audit (FAU)

FAU_GEN.1/VPN Audit Data Generation

FAU_GEN.1.1/VPN

The TSF **and [selection, choose one of: *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* ***the Auditable Events tables***].

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.

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 the Auditable Events table. 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).

The auditable events defined in the Auditable Events table are for the SFRs that are explicitly defined in this PP-Module. 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. 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/VPN

The TSF **and [selection, choose one of: *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*].

Evaluation Activities

FAU_GEN.1/VPN
TSS
The evaluator shall examine the TSS to determine that it describes the auditable events and the component that is responsible for each type of auditable event.

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

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

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

For each required auditable event, the evaluator shall examine the operational guidance to determine that it is clear to the reader where each event is generated (e.g. the TSF may generate its own audit logs in one location while the platform-provided auditable events are generated elsewhere).

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

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

FAU_SEL.1/VPN Selective Audit

FAU_SEL.1.1/VPN

The **[selection, choose one of: *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 or 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 the Auditable Events table

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

Evaluation Activities

FAU_SEL.1/VPN
TSS
There are no TSS EAs for this SFR.

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

Tests
The evaluator shall perform the following tests:

- *Test 41: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.*
- *Test 42: [conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.*

A.3 Implementation-based Requirements

A.3.1 Auditable Events for Implementation-Dependent SFRs

Table 6: Auditable Events for Implementation-based Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FDP_VPN_EXT.1	No events specified	N/A

A.3.2 User Data Protection (FDP)

FDP_VPN_EXT.1 Split Tunnel Prevention

FDP_VPN_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 implementation-based on the MDF PP being the Base-PP claimed by the TOE. In this case, this requirement must be claimed.

For all other Base-PPs, this requirement is strictly optional.

This requirement is used when the VPN 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.

Evaluation Activities

FDP_VPN_EXT.1
TSS
The evaluator shall verify that the TSS section of the ST describes the routing of IP traffic through processes on the TSF when a VPN client is enabled. The evaluator shall ensure that the description indicates which traffic does not go through the VPN and which traffic does and that a configuration exists for each baseband protocol in which only the traffic identified by the ST author is necessary for establishing the VPN connection (IKE traffic and perhaps HTTPS or DNS traffic) is not encapsulated by the VPN protocol (IPsec). The ST author shall also identify in the

TSS section any differences in the routing of IP traffic when using any supported baseband protocols (e.g. Wi-Fi, LTE).

Guidance
The evaluator shall verify that the following is addressed by the documentation:

- *The description above indicates that if a VPN client is enabled, all configurations route all IP traffic (other than IP traffic required to establish the VPN connection) through the VPN client.*
- *The guidance describes how the user or administrator can configure the TSF to meet this requirement.*

Tests
The evaluator shall perform the following test:

Step 1 - The evaluator shall use the platform to enable a network connection without using IPsec. The evaluator shall use a packet sniffing tool between the platform and an internet-connected network. The evaluator shall turn on the sniffing tool and perform actions with the device such as navigating to websites, using provided applications, accessing other internet resources (Use Case 1), accessing another VPN client (Use Case 2), or accessing an IPsec-capable network device (Use Case 3). The evaluator shall verify that the sniffing tool captures the traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 2 - The evaluator shall configure an IPsec VPN client that supports the routing specified in this requirement, and if necessary, configure the device to perform the routing specified as described in the AGD guidance. The evaluator shall turn on the sniffing tool, establish the VPN connection, and perform the same actions with the device as performed in the first step. The evaluator shall verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 3 - The evaluator shall examine the traffic from both step one and step two to verify that all IP traffic, aside from and after traffic necessary for establishing the VPN (such as IKE, DNS, and possibly HTTPS), is encapsulated by IPsec.

Step 4 - The evaluator shall attempt to send packets to the TOE outside the VPN connection and shall verify that the TOE discards them.

Appendix B - Selection-based Requirements

B.1 Auditable Events for Selection-based SFRs

Table 7: Auditable Events for Selection-based Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_EAP_EXT.1	No events specified	N/A
FIA_HOTP_EXT.1	No events specified	N/A
FIA_PSK_EXT.1	No events specified	N/A
FIA_PSK_EXT.2	No events specified	N/A
FIA_PSK_EXT.3	No events specified	N/A
FIA_PSK_EXT.4	No events specified	N/A
FIA_PSK_EXT.5	No events specified	N/A
FIA_TOTP_EXT.1	No events specified	N/A

B.2 Cryptographic Support (FCS)

FCS_EAP_EXT.1 EAP-TLS

The inclusion of this selection-based component depends upon selection in FCS_IPSEC_EXT.1.11.

FCS_EAP_EXT.1.1

The TSF shall implement [**selection**: *EAP-TLS protocol as specified in RFC 5216*, *EAP-TTLS as specified in RFC 5881*] as updated by RFC 8996 with TLS implemented using mutual authentication in accordance with the TLS functional package.

FCS_EAP_EXT.1.2

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

FCS_EAP_EXT.1.3

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

FCS_EAP_EXT.1.4

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

Evaluation Activities

FCS_EAP_EXT.1
TSS
The evaluator shall verify that the TS describes the use of EAP options for each of the selected peer authentication mechanisms, that TLS with mutual authentication is used, that the random values are from an appropriate source, and that the EAP MSK is derived from the TLS master key and is used as the IKEv2 shared key.

Guidance
The evaluator shall verify that the guidance documents describe any configurable features of the EAP or TLS functionality, including instructions for configuration of the authenticators and registration processes for clients.

Tests
Testing for TLS functionality is in accordance with the TLS package. For each supported EAP method claimed in FCS_EAP_TLS_EXT.1.1 and for each authentication method claimed in FCS_EAP_TLS_EXT.1.3, the evaluator shall perform the following tests:

- *Test 43: The evaluator shall follow AGD guidance to configure the TSF to use the EAP method claimed. The evaluator shall follow AGD guidance to configure the TSF to use the authentication method claimed and, for EAP-TTLS, register a client with the appropriate key material required for the authentication method. The evaluator shall establish an VPN session using a test client with a valid certificate and, for EAP-TTLS, configured to provide a correct value for the configured authenticator. The evaluator shall observe the the VPN session is successful.*
- *Test 44: (conditional for EAP-TTLS support): The evaluator shall cause the test client with a valid certificate to send an invalid authenticator for the claimed authentication method: For HOTP, replay the HOTP value sent previously, For TOTP or PSK, modify a byte of the properly constructed value,and observe that the TSF aborts the session*
- *Test 45: The evaluator shall establish a new, valid certificate for a test client using an identifier not corresponding to a registered user. For EAP-TTLS, the evaluator shall cause the test client using this certificate to send a correct authenticator value for the registered user. The evaluator shall initiate a VPN session from the test client to the TSF and observe that the TSF aborts the session.*
- *Test 46: The evaluator shall follow AGD guidance to configure the TSF to use a supported EAP method and register the user with the key material required for a supported authentication method. The evaluator shall configure a test client to respond to an IKEv2 exchange with EAP-request, providing valid phase 1 handshake and valid TLS handshake, but computing the phase 2 shared key using standard (non-EAP) methods. The evaluator shall initiate a VPN session between the test client and the TSF, and observe that the TSF aborts the session.*

B.3 Identification and Authentication (FIA)

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

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

The inclusion of this selection-based component depends upon selection in FIA_PSK_EXT.4.2.

FIA_HOTP_EXT.1.1

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

FIA_HOTP_EXT.1.2

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

FIA_HOTP_EXT.1.3

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

FIA_HOTP_EXT.1.4

The TSF shall use [**selection**: *SHA-1*, *SHA-256*, *SHA-384*, *SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] and message digest sizes [**selection**: *160*, *256*, *384*, *512*] to derive an HOTP hash from the HOTP seed and counter.

FIA_HOTP_EXT.1.5

The TSF shall truncate the HOTP hash per FIA_HOTP_EXT.1.4 to create an HOTP of [**selection**:

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

].

FIA_HOTP_EXT.1.6

The TSF shall [**selection**:

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

].

FIA_HOTP_EXT.1.7

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

FIA_HOTP_EXT.1.8

The TSF shall increment the counter after each successful authentication.

Application Note: The selection FIA_HOTP_EXT.1.4 must be consistent with the key size specified for the size of the keys used in conjunction with the keyed-hash message authentication.

In FIA_HOTP_EXT.1.5 the ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

In FIA_HOTP_EXT.1.6 the ST may select throttle requests, account lockout, or both.

The HOTP seed and all derived values are considered secret keys for purposes of protection.

This requirement is selection-dependent on FIA_PSK_EXT.4.

Evaluation Activities

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

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

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

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

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

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

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

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

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

Guidance
The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_HOTP_EXT.1 requirements.

Tests
The evaluator shall configure the TOE to use a supported HOTP factor then:

- *Test 47: Attempt to establish a connection using a factor from a different client. The test passes if the client fails to connect.*
- *Test 48: Attempt multiple connections outside the boundary set in FIA_HOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- *Test 49: Attempt to use an HOTP that is outside of the value allowed for resynchronization. The test passes if the client fails to connect.*
- *Test 50: Attempt to connect with a valid HOTP, disconnect and attempt to authenticate again with the same HOTP value. The test passes if the client connects the first time and fails to connect the second time. If the HOTP generated is duplicated the test may be repeated.*

FIA_PSK_EXT.1 Pre-Shared Key Composition

The inclusion of this selection-based component depends upon selection in FCS_IPSEC_EXT.1.11.

FIA_PSK_EXT.1.1

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

FIA_PSK_EXT.1.2

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

Application Note: FIA_PSK_EXT.1 includes the options for MFA solutions.

If any selection including "generated bit-based" is chosen, then FIA_PSK_EXT.2 must be included.

If any selection including Password-based keys is chosen, then FIA_PSK_EXT.3 must be included.

If any selection including HMAC-based one-time password keys is chosen, then FIA_PSK_EXT.4 must be included.

If any selection including time-based one-time password is chosen, then FIA_PSK_EXT.5 must be included.

This requirement is selection dependent on FCS_IPSEC_EXT.1.11 or inclusion of FPF_MFA_EXT.1.

Authentication options may vary between use cases, the VPN may support different options for peer to peer than for client to gateway. If this is the case the applicable selections shall be mapped to their use cases.

The first four selections are for single factor authentication options, the last four selections are for multifactor authentication options.

Evaluation Activities

FIA_PSK_EXT.1
TSS
The evaluator shall examine the TSS to ensure that it identifies all protocols that allow pre-shared keys. For each protocol identified by the requirement, the evaluator shall confirm that the TSS states which pre-shared key selections are supported.

Guidance
The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required.

Tests
The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE).

- *Test 51: For each mechanism selected in FIA_PSK_EXT.1.2, the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection.*

FIA_PSK_EXT.2 Generated Pre-Shared Keys

The inclusion of this selection-based component depends upon selection in FIA_PSK_EXT.1.2.

FIA_PSK_EXT.2.1

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

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

].

Application Note: Generated PSKs are expected to be shared between components via an out of band mechanism.

This requirement is selection-dependent on FIA_PSK_EXT.1.

Evaluation Activities

FIA_PSK_EXT.2
TSS
If "generate" is selected, the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1 and the output matches the size selected in FIA_PSK_EXT.2.1.

Guidance
The evaluator shall confirm the operational guidance contains instructions for entering generated pre-shared keys for each protocol identified in the FIA_PSK_EXT.1.1.

Tests

- *Test 52: [conditional] If generate was selected the evaluator shall generate a pre-shared key and confirm the output matches the size selected in FIA_PSK_EXT.2.1.*

FIA_PSK_EXT.3 Password-Based Pre-Shared Keys

The inclusion of this selection-based component depends upon selection in FIA_PSK_EXT.1.2.

FIA_PSK_EXT.3.1

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

FIA_PSK_EXT.3.2

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

FIA_PSK_EXT.3.3

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

FIA_PSK_EXT.3.4

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

FIA_PSK_EXT.3.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 and with entropy of [**assignment**: *value equal to or greater than 128*] bits.

FIA_PSK_EXT.3.6

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

FIA_PSK_EXT.3.7

The TSF shall [**selection**: *provide a password strength meter, check the*

password against a denylist, perform no action to assist the user in choosing a strong password].

Application Note: For FIA_PSK_EXT.3.1, the ST author assigns the maximum size of the PSK it supports; it must support at least 64 characters or a length defined by the platform.

For FIA_PSK_EXT.3.2, the ST author assigns any other supported characters; if there are no other supported characters, they should select "no other characters."

For FIA_PSK_EXT.3.3, the ST author selects the parameters based on the PBKDF used by the TSF.

For FIA_PSK_EXT.3.4 If the minimum length is settable, then the ST author chooses "a value settable by the administrator." If the minimum length is not settable, the ST author fills in the assignment with the minimum length the PSK must be. This requirement is to ensure bounds work properly.

For FIA_PSK_EXT.3.7, the ST author may select one, both, or neither of the functions in alignment with NIST SP 800-63b.

This requirement is selection-dependent on FIA_PSK_EXT.1.

Evaluation Activities

FIA_PSK_EXT.3
TSS
The evaluator shall examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are used.

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

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

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

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

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

The evaluator shall confirm that the minimum length is described.

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

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

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

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

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

Tests

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

- *Test 53: The evaluator shall compose a pre-shared key of at least 64 characters that contains a combination of the allowed characters in accordance with the FIA_PSK_EXT.1.3 and verify that a successful protocol negotiation can be performed with the key.*
- *Test 54: [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length and invalid lengths that are below the minimum length, above the maximum length, null length, empty length, or zero length. The minimum test should be successful, and the invalid lengths must be rejected by the TOE.*
- *Test 55: [conditional]: If the TOE initiates connections, initiate and establish a remote connection, disconnect from the connection, verify that the PSK is required when initiating the connection a second time.*
- *Test 56: [conditional]: If the TOE supports a password meter, the evaluator shall enter a password and verify the password checker responds per the description in the TSS.*
- *Test 57: [conditional]: If the TOE supports a password denylist, the evaluator shall enter a denylisted password and verify that the password is rejected or flagged as such.*

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

The inclusion of this selection-based component depends upon selection in FIA_PSK_EXT.1.2.

FIA_PSK_EXT.4.1

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

FIA_PSK_EXT.4.2

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

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

This requirement is selection-dependent on FIA_PSK_EXT.1

The selection "verify the HOTP via an external authentication server" is intended to cover the case where the TOE is not doing the verifying, such as if the when the VPN GW or an authentication server fulfills that function. If a client supports this setup for a peer to peer use case then the "verify the HOTP" selection may be included.

Evaluation Activities

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

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

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

Tests

- *Test 58: The evaluator shall configure the TOE to use a supported HOTP factor, then attempt to establish a connection using that factor. The evaluator shall verify the client prompts the user for the HOTP before initiating the connection. The evaluator shall verify the server validates the HOTP or receives confirmation from an authentication server before establishing the channel.*

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

The inclusion of this selection-based component depends upon selection in

***FIA_PSK_EXT.1.2*.**

FIA_PSK_EXT.5.1

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

FIA_PSK_EXT.5.2

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

Application Note: If verify the TOTP is selected then FIA_TOTP_EXT.1 must be included.

This requirement is dependent on FIA_PSK_EXT.1.

The selection "verify the TOTP via an external authentication server" is intended to cover the case where the TOE is not doing the verifying, such as if the when the VPN GW or an authentication server fulfills that function. If a client supports this setup for a peer to peer use case then the "verify the TOTP" selection may be included.

Evaluation Activities

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

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

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

Tests

- *Test 59: The evaluator shall configure the TOE to use a supported TOTP factor, then attempt to establish a connection using that factor. The evaluator shall verify the client prompts the user for the TOTP before initiating the connection. The evaluator shall verify the server validates the TOTP or receives confirmation from an authentication server before establishing the channel.*

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

***The inclusion of this selection-based component depends upon selection in FIA_PSK_EXT.5.2*.**

FIA_TOTP_EXT.1.1

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

FIA_TOTP_EXT.1.2

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

FIA_TOTP_EXT.1.3

The TSF shall generate a new TOTP seed for each client.

FIA_TOTP_EXT.1.4

The TSF shall use [**selection**: *SHA-1*, *SHA-256*, *SHA-384*, *SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] and message digest sizes [**selection**: *160*, *256*, *384*, *512*] to derive a TOTP hash from the TOTP seed and current time provided by NTP.

FIA_TOTP_EXT.1.5

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

- *administrator configurable character length of at least 6*

- *non-configurable character length of [**selection, choose one of**: 6, 7, 8, 9, 10]*

].

FIA_TOTP_EXT.1.6

The TSF shall [**selection**:

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

].

FIA_TOTP_EXT.1.7

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

FIA_TOTP_EXT.1.8

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

FIA_TOTP_EXT.1.9

The TSF shall [**selection, choose one of**: *allow resynchronization by recording time drift within the limit of FIA_TOTP_EXT.1.8, not permit resynchronization*].

Application Note: The selection FIA_TOTP_EXT.1.4 must be consistent with the key size specified for the size of the keys used in conjunction with the keyed-hash message authentication.

In FIA_TOTP_EXT.1.5 the ST author may either provide a configurable character length of at least 6 or a preset size between 6 and 10.

In FIA_TOTP_EXT.1.6 the ST author may select throttle requests, account lockout, or both.

The TOTP seed and all derived values are considered secret keys for purposes of protection.

This requirement is selection-dependent on FIA_PSK_EXT.5.

Evaluation Activities

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

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

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

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

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

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

The evaluator shall confirm the TSS describes how the TOE handles multiple incoming requests and verify it provides an anti-hammer mechanism that matches the selections made in FIA_TOTP_EXT.1.6.

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

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

Guidance
The evaluator shall verify the operational guidance contains all configuration guidance for setting any administrative value that is configurable in the FIA_TOTP_EXT.1 requirements.

Tests
The evaluator shall configure the TOE to use a supported TOTP factor then:

- *Test 60: Attempt to establish a connection using a factor from a different client. The test passes if the client fails to connect.*
- *Test 61: Attempt multiple connections outside the boundary set in FIA_TOTP_EXT.1.6 and verify the remediation is triggered. The test passes if remediation is triggered as defined in the selections and assignments.*
- *Test 62: Attempt to use a TOTP that is outside of the value allowed for resynchronization. The test passes if the client fails to connect.*
  Attempt to connect with a valid TOTP, disconnect and attempt to authenticate again with the same TOTP. The test passes if the client connects the first time and fails to connect the second time. If the TOTP generated is duplicated the test may be repeated.

Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the PP-Module.

C.1 Extended Components Table

All extended components specified in the PP-Module are listed in this table:

Table 8: Extended Component Definitions

Functional Class	Functional Components
Cryptographic Support (FCS)	FCS_CKM_EXT Cryptographic Key Management FCS_EAP_EXT EAP-TLS FCS_IPSEC_EXT IPsec
Identification and Authentication (FIA)	FIA_BMA_EXT Biometric Activation FIA_HOTP_EXT HMAC-Based One-Time Password Pre-Shared Keys FIA_PSK_EXT Pre-Shared Key Composition FIA_TOTP_EXT Time-Based One-Time Password Pre-Shared Keys FIA_X509_EXT X.509 Certificate Use and Management
Packet Filtering (FPF)	FPF_MFA_EXT Multifactor Authentication Filtering
Protection of the TSF (FPT)	FPT_TST_EXT TSF Self-Test
User Data Protection (FDP)	FDP_VPN_EXT Subset Information Flow Control

C.2 Extended Component Definitions

C.2.1 Cryptographic Support (FCS)

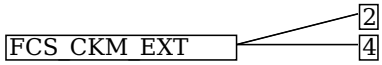
This PP-Module defines the following extended components as part of the FCS class originally defined by CC Part 2:

C.2.1.1 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, requires the TSF to securely store key data when not in use.

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

Management: FCS_CKM_EXT.2

No specific management functions are identified.

Audit: FCS_CKM_EXT.2

There are no auditable 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, choose one of**: *VPN client*, *OS*] shall store persistent secrets and private keys when not in use in OS-provided key storage.

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, choose one of**: *TOE*, *TOE platform*] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

C.2.1.2 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 the TOE
- 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 TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2

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

FCS_IPSEC_EXT.1.3

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

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [*AES-GCM-128, 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 TSF 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 8784, RFC 8247, and [**selection**: no other RFCs for hash functions, RFC 4868 for hash functions]*

].

FCS_IPSEC_EXT.1.6

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

FCS_IPSEC_EXT.1.7

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

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

]. 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 TSF shall ensure that IKE protocols implement DH Groups

- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and**

[**selection**:

- *[**selection**: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526*
- *[**selection**: 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS, **no other DH Groups**] according to RFC 5114*

].

FCS_IPSEC_EXT.1.9

The TSF shall generate the secret value x used in the IKE DH 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) numbers of bits that is at least twice the “bits of security” value associated with the negotiated DH group as listed in Table 2 of NIST SP 800-57, Recommendation for Key Management – Part 1: General*] bits.

FCS_IPSEC_EXT.1.10

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

FCS_IPSEC_EXT.1.11

The TSF shall ensure that all IKE protocols perform peer authentication using a [**selection**: *RSA*, *ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: *Pre-shared keys*, *Pre-shared Keys transmitted via EAP-TLS*, *Pre-shared Keys transmitted via EAP-TTLS with mutual authentication*, *no other method*].

FCS_IPSEC_EXT.1.12

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

FCS_IPSEC_EXT.1.13

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

FCS_IPSEC_EXT.1.14

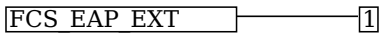
The [**selection**: *TSF*, *VPN Gateway*] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [**selection**: *IKEv1 Phase 1*, *IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric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

C.2.1.3 FCS_EAP_EXT EAP-TLS

Family Behavior

Components in this family describe the requirements for EAP-TLS.

Component Leveling



FCS_EAP_EXT.1, EAP-TLS, defines the use of EAP-TLS.

Management: FCS_EAP_EXT.1

No specific management functions are identified.

Audit: FCS_EAP_EXT.1

No specific audit functions are identified.

FCS_EAP_EXT.1 EAP-TLS

Hierarchical to: No other components.

Dependencies to: FCS_IPSEC_EXT.1 IPsec

FCS_EAP_EXT.1.1

The TSF shall implement [**selection**: *EAP-TLS protocol as specified in RFC 5216*, *EAP-TTLS as specified in RFC 5881*] as updated by RFC 8996 with TLS implemented using mutual authentication in accordance with the TLS functional package.

FCS_EAP_EXT.1.2

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

FCS_EAP_EXT.1.3

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

FCS_EAP_EXT.1.4

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

C.2.2 Identification and Authentication (FIA)

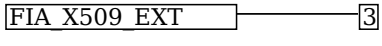
This PP-Module defines the following extended components as part of the FIA class originally defined by CC Part 2:

C.2.2.1 FIA_X509_EXT X.509 Certificate Use and Management

Family Behavior

Components in this family describe the requirements that pertain to IP 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 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 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, choose one of**: *VPN client*, *OS*] shall [**selection, choose one of**: *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

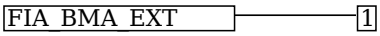
The [**selection, choose one of**: *VPN client*, *OS*] shall not establish an SA if a certificate or certificate path is deemed invalid.

C.2.2.2 FIA_BMA_EXT Biometric Activation

Family Behavior

Components in this family describe the requirements for biometrics when using the VPN client.

Component Leveling



FIA_BMA_EXT.1, Biometric Activation, defines the use of biometrics when using the VPN client.

Management: FIA_BMA_EXT.1

No specific management functions are identified.

Audit: FIA_BMA_EXT.1

No specific audit functions are identified.

FIA_BMA_EXT.1 Biometric Activation

Hierarchical to: No other components.

Dependencies to: No dependencies.

FIA_BMA_EXT.1.1

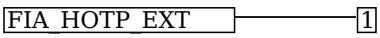
The TSF shall leverage the platform biometric features to confirm the user before initiating a trusted channel.

C.2.2.3 FIA_HOTP_EXT HMAC-Based One-Time Password Pre-Shared Keys

Family Behavior

Components in this family define requirements for use of HMAC-Based One-Time password authentication, including generation methods and usage restrictions.

Component Leveling



FIA_HOTP_EXT.1, HMAC-Based One-Time Password Pre-Shared Keys, defines the implementation of HOTP.

Management: FIA_HOTP_EXT.1

No specific management functions are identified.

Audit: FIA_HOTP_EXT.1

No specific audit functions are identified.

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

Hierarchical to: No other components.

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

FIA_HOTP_EXT.1.1

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

FIA_HOTP_EXT.1.2

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

FIA_HOTP_EXT.1.3

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

FIA_HOTP_EXT.1.4

The TSF shall use [**selection**: *SHA-1*, *SHA-256*, *SHA-384*, *SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] and message digest sizes [**selection**: *160*, *256*, *384*, *512*] to derive an HOTP hash from the HOTP seed and counter.

FIA_HOTP_EXT.1.5

The TSF shall truncate the HOTP hash per FIA_HOTP_EXT.1.4 to create an HOTP of [**selection**:

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

].

FIA_HOTP_EXT.1.6

The TSF shall [**selection**:

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

].

FIA_HOTP_EXT.1.7

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

FIA_HOTP_EXT.1.8

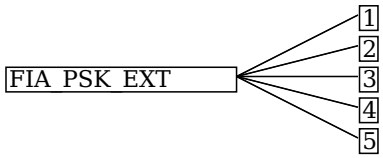
The TSF shall increment the counter after each successful authentication.

C.2.2.4 FIA_PSK_EXT Pre-Shared Key Composition

Family Behavior

Components in this family describe 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.

FIA_PSK_EXT.2, Generated Pre-Shared Keys, defines the use and composition of generated pre-shared keys used for IPsec.

FIA_PSK_EXT.3, Password-Based Pre-Shared Keys, defines the use and composition of password-based pre-shared keys used for IPsec.

FIA_PSK_EXT.4, HMAC-Based One-Time Password Pre-shared Keys Support, defines the use and composition of HOTP pre-shared keys used for IPsec.

FIA_PSK_EXT.5, Time-Based One-Time Password Pre-shared Keys Support, defines the use and composition of TOTP pre-shared keys used for IPsec.

Management: FIA_PSK_EXT.1

No specific management functions are identified.

Audit: FIA_PSK_EXT.1

No specific audit functions are identified.

FIA_PSK_EXT.1 Pre-Shared Key Composition

Hierarchical to: No other components.

Dependencies to: FCS_IPSEC_EXT.1 IPsec

FIA_PSK_EXT.1.1

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

FIA_PSK_EXT.1.2

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

Management: FIA_PSK_EXT.2

No specific management functions are identified.

Audit: FIA_PSK_EXT.2

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.2 Generated Pre-Shared Keys

Hierarchical to: No other components.

Dependencies to: FIA_PSK_EXT.1

FIA_PSK_EXT.2.1

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

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

].

Management: FIA_PSK_EXT.3

No specific management functions are identified.

Audit: FIA_PSK_EXT.3

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.3 Password-Based Pre-Shared Keys

Hierarchical to: No other components.

Dependencies to: FIA_PSK_EXT.1

FIA_PSK_EXT.3.1

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

FIA_PSK_EXT.3.2

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

FIA_PSK_EXT.3.3

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

FIA_PSK_EXT.3.4

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

FIA_PSK_EXT.3.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 and with entropy of [**assignment**: *value equal to or greater than 128*] bits.

FIA_PSK_EXT.3.6

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

FIA_PSK_EXT.3.7

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

Management: FIA_PSK_EXT.4

No specific management functions are identified.

Audit: FIA_PSK_EXT.4

No specific audit functions are identified.

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

Hierarchical to: No other components.

Dependencies to: FIA_PSK_EXT.1

FIA_PSK_EXT.4.1

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

FIA_PSK_EXT.4.2

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

Management: FIA_PSK_EXT.5

No specific management functions are identified.

Audit: FIA_PSK_EXT.5

No specific audit functions are identified.

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

Hierarchical to: No other components.

Dependencies to: FIA_PSK_EXT.1

FIA_PSK_EXT.5.1

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

FIA_PSK_EXT.5.2

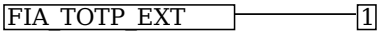
The TSF shall [**selection, choose one of**: *verify the TOTP, verify the TOTP via an external authentication server*] before establishing an incoming connection.

C.2.2.5 FIA_TOTP_EXT Time-Based One-Time Password Pre-Shared Keys

Family Behavior

Components in this family define requirements for use of Time-Based One-Time password authentication, including generation methods and usage restrictions.

Component Leveling



FIA_TOTP_EXT.1, Time-Based One-Time Password Pre-Shared Keys, defines the implementation of TOTP.

Management: FIA_TOTP_EXT.1

No specific management functions are identified.

Audit: FIA_TOTP_EXT.1

No specific audit functions are identified.

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

Hierarchical to: No other components.

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

FIA_TOTP_EXT.1.1

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

FIA_TOTP_EXT.1.2

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

FIA_TOTP_EXT.1.3

The TSF shall generate a new TOTP seed for each client.

FIA_TOTP_EXT.1.4

The TSF shall use [**selection**: *SHA-1*, *SHA-256*, *SHA-384*, *SHA-512*] with key sizes [**assignment**: *key size (in bits) used in HMAC*] and message digest sizes [**selection**: *160*, *256*, *384*, *512*] to derive a TOTP hash from the TOTP seed and current time provided by NTP.

FIA_TOTP_EXT.1.5

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

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

].

FIA_TOTP_EXT.1.6

The TSF shall [**selection**:

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

].

FIA_TOTP_EXT.1.7

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

FIA_TOTP_EXT.1.8

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

FIA_TOTP_EXT.1.9

The TSF shall [**selection, choose one of**: *allow resynchronization by recording time drift within the limit of FIA_TOTP_EXT.1.8*, *not permit resynchronization*].

C.2.3 Packet Filtering (FPF)

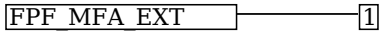
This class contains families that describe packet filtering behavior. Packet filtering refers to the notion that network traffic that is transmitted “through” the TOE (i.e. the source and destination of the traffic is not the TOE but the TOE is on the routing path between these two entities) can be treated differently by the TSF based on attributes associated with the traffic. As this class is defined solely to contain an extended component defined for this PP-Module, it has one family, FPF_MFA_EXT.

C.2.3.1 FPF_MFA_EXT Multifactor Authentication Filtering

Family Behavior

Components in this family describe the requirements for multifactor authentication filtering when using the VPN client.

Component Leveling



FPF_MFA_EXT.1, Multifactor Authentication Filtering, defines the use and composition of multifactor authentication filtering.

Management: FPF_MFA_EXT.1

No specific management functions are identified.

Audit: FPF_MFA_EXT.1

No specific audit functions are identified.

FPF_MFA_EXT.1 Multifactor Authentication Filtering

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPF_MFA_EXT.1.1

The TSF shall not forward packets to the internal network until the IKE/IPsec tunnel has been established, except those necessary to ensure that the client is authenticated according to FIA_PSK_EXT.1.

C.2.4 Protection of the TSF (FPT)

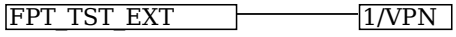
This PP-Module defines the following extended components as part of the FPT class originally defined by CC Part 2:

C.2.4.1 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/VPN, 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/VPN

No specific management functions are identified.

Audit: FPT_TST_EXT.1/VPN

There are no auditable events foreseen.

FPT_TST_EXT.1/VPN TSF Self-Test

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPT_TST_EXT.1.1/VPN

The [**selection, choose one of**: *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/VPN

The [**selection, choose one of**: *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 OE*].

C.2.5 User Data Protection (FDP)

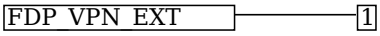
This PP-Module defines the following extended components as part of the FDP class originally defined by CC Part 2:

C.2.5.1 FDP_VPN_EXT Subset Information Flow Control

Family Behavior

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

Component Leveling



FDP_VPN_EXT.1, Split Tunnel Prevention, requires the TSF to process all IP traffic through its VPN client functionality.

Management: FDP_VPN_EXT.1

No specific management functions are identified.

Audit: FDP_VPN_EXT.1

There are no auditable events foreseen.

FDP_VPN_EXT.1 Split Tunnel Prevention

Hierarchical to: No other components.

Dependencies to: FCS_IPSEC_EXT.1 IPsec

FDP_VPN_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 D - Implicitly Satisfied Requirements

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this PP-Module. These requirements are not featured explicitly as SFRs and should not be included in the ST. They are not included as standalone SFRs because it would increase the time, cost, and complexity of evaluation. This approach is permitted by [CC] Part 1, 8.2 Dependencies between components.

This information benefits systems engineering activities which call for inclusion of particular security controls. Evaluation against the PP-Module provides evidence that these controls are present and have been evaluated.

Table 9: Implicitly Satisfied Requirements

Requirement	Rationale for Satisfaction
FCS_CKM.2 – Cryptographic Key Distribution, or FCS_COP.1 – Cryptographic Operation	FCS_CKM.1 (which is defined in this PP-Module as FCS_CKM.1/VPN) requires one of FCS_CKM.2 or FCS_COP.1 to be claimed so that the generated keys can serve some security-relevant purpose. Each of the Base-PPs for this PP-Module define an iteration of FCS_COP.1 for symmetric cryptography that is expected to use the IKE keys generated by FCS_CKM.1/VPN. Therefore, this dependency is satisfied through requirements defined in the Base-PPs.
FCS_CKM.4 – Cryptographic Key Destruction	FCS_CKM.1 (which is defined in this PP-Module as FCS_CKM.1/VPN) requires FCS_CKM.4 to be claimed so that the generated keys are not disclosed through improper or nonexistent key destruction methods. Each of the supported Base-PPs except for the App PP define FCS_CKM_EXT.4 as an extended SFR, which defines key destruction functionality consistent with FCS_CKM.4, but with additional details that are specific to the respective technology types of the Base-PP. When the App PP is the Base-PP, this PP-Module defines its own instance of FCS_CKM_EXT.4 to achieve the same purpose. The dependency on FCS_CKM.4 is considered to be satisfied through the fact that a compliant TOE will always claim FCS_CKM_EXT.4, which is intended to satisfy the same purpose.
FCS_COP.1 – Cryptographic Operation	FCS_IPSEC_EXT.1 has a dependency on FCS_COP.1 because of the cryptographic operations that are needed in support of implementing the IPsec protocol. FCS_COP.1 is not defined in this PP-Module because each of the supported Base-PPs define iterations of FCS_COP.1 that support the functions that are relevant to IPsec.
FMT_MTD.1 – Management of TSF Data	FAU_SEL.1/VPN has a dependency on FMT_MTD.1 to enforce appropriate access controls on the audit configuration, as this is TSF data. This SFR is not explicitly defined in any of the supported Base-PPs but the dependency is implicitly addressed by each Base-PP in the following manner: • GPOS PP: The GPOS PP implicitly defines the existence of ‘user’ and ‘administrator’ roles in the extended SFRs FMT_MOF_EXT.1 and FMT_SMF_EXT.1. A TOE that conforms to this Base-PP can associate the ability to perform the functionality defined by FAU_SEL.1/VPN to one or both of these roles. • MDF PP: The MDF PP implicitly defines the existence of ‘user,’ ‘administrator,’ and ‘MDM’ roles in the extended SFRs FMT_MOF_EXT.1 and FMT_SMF_EXT.1. A TOE that conforms to this Base-PP can associate the ability to perform the functionality defined by FAU_SEL.1/VPN to one or more of these roles. • App PP: The App PP does not define the existence of a separately authenticated management interface; instead, the App PP assumes that authentication to the underlying OS platform is sufficient authorization to access the application’s management functionality. • MDM PP: The MDM PP defines the existence of management roles in FMT_SMR.1(1). A TOE that conforms to this Base-PP can associate the ability to perform the functionality defined by FAU_SEL.1/VPN to one or more of the roles defined here.
FPT_STM.1 – Reliable Time Stamps	FAU_GEN.1/VPN has a dependency on FPT_STM.1 because audit records are required to have timestamps that are based on reliable clock data. All of the supported Base-PPs either define this requirement explicitly or provide rationale for why the reader should expect that a reliable clock service should be present. Depending on the claimed Base-PP, the dependency is satisfied in the following manner: • GPOS PP: The GPOS PP states that FPT_STM.1 is implicitly satisfied by the requirements of FAU_GEN.1 since that requirement could not be satisfied if no clock service was present. Additionally, a clock service is reasonably assumed to be provided by a general-purpose OS. • MDF PP: The MDF PP explicitly defines FPT_STM.1. • App PP: The App PP assumption A.PLATFORM assumes that the general-purpose

	computing platform on which the TOE is installed is ‘a trustworthy computing platform.’ System time data is not explicitly mentioned but a clock service is reasonably assumed to be provided by a general-purpose computer. • MDM PP: The MDM PP assumption A.MDM_SERVER_PLATFORM assumes that the platform on which the TOE is installed will provide reliable time services.
FPT_STM.1 – Reliable Time Stamps	FAU_GEN.1 has a dependency on FPT_STM.1. While not explicitly stated in the PP, it is assumed that this will be provided by the underlying hardware platform on which the TOE is installed. This is because the TOE is installed as a software or firmware product that runs on general-purpose computing hardware so a hardware clock is assumed to be available.
FPT_STM.1 – Reliable Time Stamps	FIA_X509_EXT.1 has a dependency on FPT_STM.1. While not explicitly stated in the PP, it is assumed that this will be provided by the underlying hardware platform on which the TOE is installed. This is because the TOE is installed as a software or firmware product that runs on general-purpose computing hardware so a hardware clock is assumed to be available.

Appendix E - Entropy Documentation and Assessment

The TOE does not require any additional supplementary information to describe its entropy sources beyond the requirements outlined in the Base-PPs. As with other Base-PP requirements, the only additional requirement is that the entropy documentation also applies to the specific VPN client capabilities of the TOE in addition to the functionality required by the claimed Base-PP.

Appendix F - Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
Base-PP	Base Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
cPP	Collaborative Protection Profile
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DH	Diffie-Hellman
DN	Distinguished Name
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
EP	Extended Package
ESP	Encapsulating Security Protocol
EUD	End-User Device
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
FP	Functional Package
FQDN	Fully Qualified Domain Name
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
OE	Operational Environment
OS	(General Purpose) Operating System
OSP	Organizational Security Policy
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
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
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
VPN	Virtual Private Network

Appendix G - 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.
[App PP]	Protection Profile for Application Software, Version 1.4, October 2021
[GPOS 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
[MDM PP]	Protection Profile for Mobile Device Management, Version 4.0, April 2019
[SD]	Supporting Document Mandatory Technical Document, PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, November 2019