Comment: Comment-1-

# PP-Module for VPN Client

Version: 2.6
2025-01-31
**National Information Assurance Partnership**

## Revision History

| Version | Date | Comment |
|---|---|---|
| 2.6 | 2025-01-31 | CC:2022 conversion, limitation of cryptographic algorithms to CNSA 1.0, incorporation of TDs |
| 2.5 | 2024-06-24 | Incorporation of TC feedback:<br>• Incorporation of TDs: 0662, 0672, 0690, 0697, 0711, 0725, 0753, 0788<br>• Corrections to Base-PP references<br>• Definition of auditable events for Additional SFRs<br>• Explicit association of evaluation activities with components and elements |
| 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

FIA_X509_EXT references to the Base-PPs are now removed and where appropriate the X.509 package is referenced instead. However, it's unclear whether there is still sufficient mechanism to actually 'force' the X.509 SFRs to be included. That is to say, there is nothing in here that says "because IPsec functionality is dependent on X.509 validation, and because the Base-PPs conform to the X.509 FP, the ST shall make the relevant X.509 FP claims." 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 4.3
- Protection Profile for Mobile Device Fundamentals (MDF PP), Version 3.3
- Protection Profile for Application Software (App PP), Version 2.0
- 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 through 5.4 of this PP-Module describe the relevant functionality for each Base-PP, including specific selections and 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- | A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module. |

| | |
|---|---|
| Configuration) | |
| Protection Profile Module (PP-Module) | An implementation-independent statement of security needs for a TOE type complementary to one or more Base-PPs. |
| 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 such as 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) that 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 shall use 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) 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.

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, and 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.

To show that the TSF implements the RFCs correctly, the evaluator shall perform the 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 what 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 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.

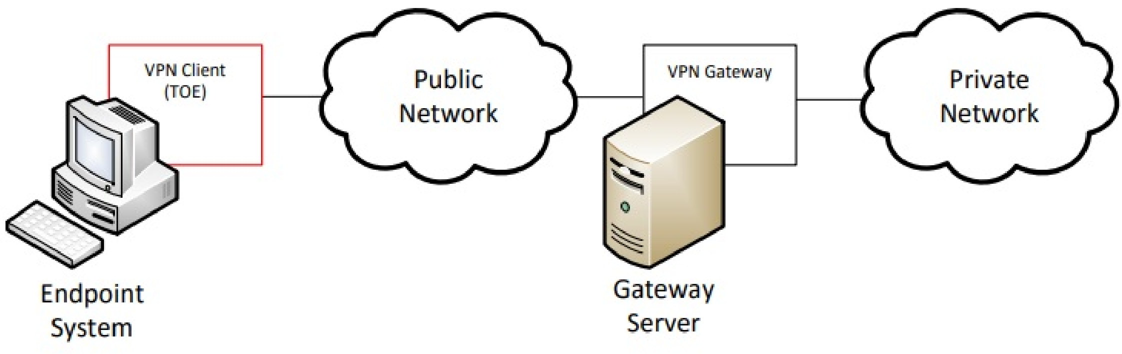



**Figure 1: TOE to VPN Gateway**

**[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 to facilitate point-to-point communications with 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.

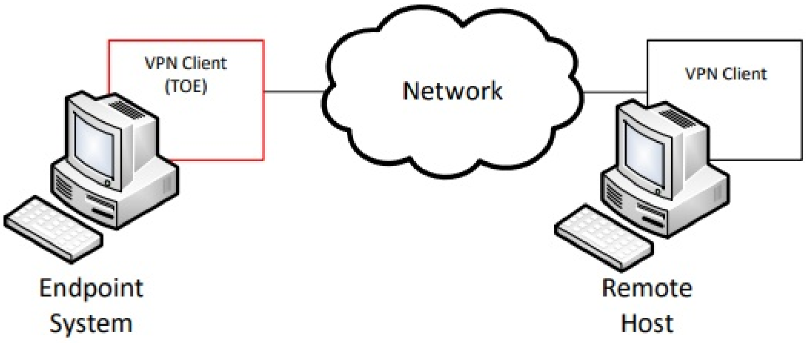



**Figure 2: TOE to VPN Client**

**[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 an SSH connection might 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.

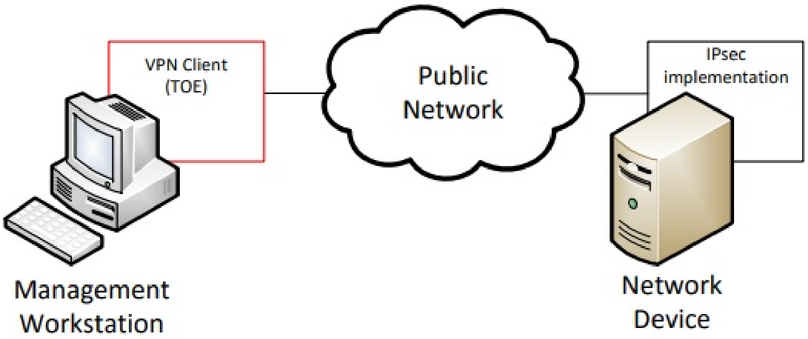



**Figure 3: TOE to IPsec-Capable Network Device**

## 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 to either 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

An ST must claim exact conformance to this PP-Module.

The evaluation methods used for evaluating the TOE are a combination of the workunits defined in [CEM] as well as the Evaluation Activities for ensuring that individual SFRs and SARs have a sufficient level of supporting evidence in the Security Target and guidance documentation and have been sufficiently tested by the laboratory as part of completing ATE_IND.1. Any functional packages this PP claims similarly contain their own Evaluation Activities that are used in this same manner.

### CC Conformance Claims

This PP-Module is conformant to Part 2 (extended) and Part 3 (extended) of Common Criteria CC:2022, Revision 1.

### PP Claim

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

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

- Protection Profile for General Purpose Operating Systems, Version 4.3
- Protection Profile for Mobile Device Fundamentals, Version 3.3
- Protection Profile for Mobile Device Management, Version 4.0
- Protection Profile for Application Software, Version 2.0
- cPP-Module for Wireless LAN Clients, version 1.1
- PP-Module for Bluetooth, version 1.1
- PP-Module for Mobile Device Management Agent, version 1.2
- cPP-Module for Biometric Enrolment and Verification, version 1.1

### Package Claim

- This PP-Module is Functional Package for Transport Layer Security Version 2.1 conformant.
- This PP-Module is Functional Package for X.509 Version 1.0 conformant.
- This PP-Module is Assurance Package for Flaw Remediation Version 1.0 conformant.

The functional packages to which the PP conforms may include SFRs that are not mandatory to claim for the sake of conformance. An ST that claims one or more of these functional packages may include any non-mandatory SFRs that are appropriate to claim based on the capabilities of the TSF and on any triggers for their inclusion based inherently on the SFR selections made.

# 3 Security Problem Definition

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.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 incorrect 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.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.

**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-middletype 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.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 reused in sending network traffic to a user other than that intended by the sender of the original network traffic.

## 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 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.

## 4.2 Security Objectives Rationale

This section describes how the assumptions and organizational security policies map to operational environment security objectives.

**Table 1: Security Objectives Rationale**

| Assumption or OSP | Security Objectives | Rationale |
|---|---|---|
| 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 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 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.

### 5.1.1 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 186-5, "Digital Signature Standard (DSS)," Appendix B.3 for RSA schemes***
- ***FIPS PUB 186-5, "Digital Signature Standard (DSS)," Appendix B.4 for ECDSA schemes and implementing "NIST curve" P-384 and no other curves.***

**]** and specified cryptographic key sizes [*equivalent to, or greater than, a symmetric key strength of 192 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.

#### 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 permissible 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-256 as specified in RFC 4106*, [**selection**: *AES-CBC-256 (both specified by RFC 3602) 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 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-256 as specified in RFC 6379 and* [**selection**: *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 [**selection**:*
    - *can be configured by [**selection**: an administrator, a VPN gateway] based on [**selection**: number of packets/number of bytes, length of time]*
    - *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 eight 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

- 20 (384-bit Random ECP) according to RFC 5114 and

[**selection**:

- *15 (3072-bit MODP)*
- *16 (4096-bit MODP)*
- *17 (6144-bit MODP)*
- *18 (8192-bit MODP)*

] according to RFC 3526.

**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 they are inserted directly into the assignment). For example, suppose the implementation supports DH group 15 (3072-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 128, and for group 20 it is 192. For

FCS_IPSEC_EXT.1.9, then, the assignment would read "[256, 384]" and for FCS_IPSEC_EXT.1.10 it would read "[128, 192]"

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.1 (or FCS_RBG_EXT.1 in the case of , 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 [**selection**:

- *IKEv1 performs peer authentication using [**selection**: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945.*
- *IKEv2 performs peer authentication using [**selection**: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: Pre-shared Keys that conform to RFC 8784, Pre-shared Keys transmitted via EAP-TTLS, EAP-TLS, 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. X.509 certificates will be validated against FIA_X509_EXT.1 from Functional Package for X.509, version 1.0, which is referenced in each supported Base-PP.

If a selection with "EAP-TLS" or "EAP-TTLS" is chosen, the selection-based requirement FCS_EAP_EXT.1 must be claimed. When an EAP method is used, verification occurs via an external authentication server.

If any selection including "pre-shared keys" is chosen, the selection-based requirement FIA_PSK_EXT.1 must be claimed. Since certificates are required for authentication, IKEv1 does not support PSKs, as PSK-based authentication and certificate-based authentication are mutually exclusive in IKEv1. Note that IKEv1 will be removed in a future version of this document.

Multifactor support can be achieved via traffic filtering in accordance with FPF_MFA_EXT.1.

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

This SFR is modified from its definition in the Base-PP by adding new selections for IKE versions and authentication methods.

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. 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.

### 5.1.2 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.

### 5.1.3 Security Management (FMT)

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

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

FMT_SMF.1.1/VPN

The TSF shall be capable of performing the following management functions: [**selection**:

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

]

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

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

Selections appropriate for the use 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.

### 5.1.4 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.

## 5.2 TOE Security Functional Requirements Rationale

The following rationale provides justification for each SFR for the TOE, showing that the SFRs are suitable to address the specified threats:

**Table 2: SFR Rationale**

| Threat | Addressed by | Rationale |
|---|---|---|
| T.TSF_CONFIGURATION | FMT_SMF.1/VPN | This SFR mitigates the threat by requiring the TOE to implement certain administratively-configurable functions. |
| | FPT_TST_EXT.1/VPN | This SFR mitigates the threat by requiring the TOE to execute self-tests that demonstrate that its integrity is maintained. |
| T.TSF_FAILURE | FPT_TST_EXT.1/VPN | This SFR mitigates the threat by requiring the TOE to execute self-tests that demonstrate that its integrity is maintained. |
| T.UNAUTHORIZED_ACCESS | FCS_EAP_EXT.1 (selection-based) | This SFR mitigates the threat by optionally implementing EAP-TLS or EAP-TTLS as a mechanism for authentication. |
| | FCS_IPSEC_EXT.1 | This SFR mitigates the threat 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 mitigates the threat by optionally defining the TOE's support for a platform-based biometric mechanism to use as an authentication mechanism. |
| | FIA_PSK_EXT.1 (selection-based) | This SFR mitigates the threat by optionally requiring support for pre-shared keys as an alternate authentication method for IPsec. |
| | FIA_PSK_EXT.2 (selection-based) | This SFR mitigates the threat 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 mitigates the threat by optionally defining the composition and use of password-based pre-shared keys used for authentication. |
| | FIA_PSK_EXT.4 (selection-based) | This SFR mitigates the threat by optionally defining HOTP as an authentication mechanism. |
| | FIA_PSK_EXT.5 (selection-based) | This SFR mitigates the threat by optionally defining TOTP as an authentication mechanism. |
| | FPF_MFA_EXT.1 (optional) | This SFR mitigates the threat by optionally enforcing a multifactor authentication requirement on an IPsec connection. |
| T.USER_DATA_ REUSE | FDP_RIP.2 | This SFR mitigates the threat by requiring the TOE or its platform to ensure that residual data is purged from the system. |
| | FPF_MFA_EXT.1 (optional) | This SFR mitigates the threat by optionally requiring the TOE to prohibit transmission of packet data aside from those packets needed to perform multifactor authentication. |

# 6 Consistency Rationale

# Appendix A - Optional SFRs

## A.1 Strictly Optional Requirements

### A.1.1 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.

### A.1.2 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.

## A.2 Objective Requirements

### A.2.1 Security Audit (FAU)

#### 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.

## A.3 Implementation-dependent Requirements

### A.3.1 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 audit data of the following auditable events:

1. Start-up and shutdown of the audit functions;
2. All auditable events for the [*not specified*] level of audit;
3. All administrative actions;
4. [*Specifically defined auditable events listed in the Auditable Events tables*].

**Application Note:** This requirement is implementation-dependent on the MDF

PP, GPOS PP, or MDM PP being the Base-PP claimed by the TOE. In this case, this requirement must be claimed.

For the App PP Base-PP, this requirement is strictly optional.

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 (, , , , and ). For any SFRs that are included as part of the TOE based on the claimed Base-PP (as defined in the Auditable Events tables in the Additional SFRs section for the corresponding Base-PP claim), 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 the audit data at least the following information:

1. Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
2. 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 the Auditable Events tables*].

# Appendix B - Selection-based Requirements

## B.1 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 support [**selection**: *EAP-TLS as specified in RFC 5216 and updated by RFC 8996*, *EAP-TTLS as specified in RFC 5281 and updated by RFC 8996*] over a protected channel **per the Base-PP** with an authentication server.

FCS_EAP_EXT.1.2

The TSF shall implement [**selection**: *EAP-TLS*, *EAP-TTLS*] with the TSF as the EAP client, an external authentication server as the EAP server and the VPN peer as the supplicant.

FCS_EAP_EXT.1.3

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

## B.2 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_PSK_EXT.1 Pre-Shared Key Composition

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

***This component must be included in the ST if any of the following SFRs are included:***

- ***FPF_MFA_EXT.1***

FIA_PSK_EXT.1.1

The TSF shall be able to use pre-shared keys for [**selection**: *IKEv2*, *multifactor authentication filtering*].

FIA_PSK_EXT.1.2

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

**Application Note:** If “pre-shared keys that conform to RFC 8784” is selected in FCS_IPSEC_EXT.1.11, a generated, bit-based PSK must be used.

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-based on FCS_IPSEC_EXT.1.11 or inclusion of FPF_MFA_EXT.1.

#### 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 256 bit-based pre-shared keys via the random number generator used by the TSF.*].

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

This requirement is selection-based on FIA_PSK_EXT.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-SHA-384*], with [**assignment**: *positive integer of 4096 or more*] iterations, and output cryptographic key sizes [*256 bits*] 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 **[selection: *FCS_RBG.1, FCS_RBG_EXT.1*]** and with entropy of [**assignment**: *value equal to or greater than 256*] bits.

**Application Note:** For the first selection, the ST author selects FCS_RBG.1 if the TOE implements its own DRBG. The ST author selects FCS_RBG_EXT.1 if is the Base-PP for the TOE and the TSF relies on a DRBG in its operational environment.

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-based on FIA_PSK_EXT.1.

#### 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.

**Application Note:** This requirement is selection-based on FIA_PSK_EXT.1

#### 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.

**Application Note:** This requirement is selection-based on FIA_PSK_EXT.1.

# 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 3: Extended Component Definitions**

| Functional Class | Functional Components |
|---|---|
| Cryptographic Support (FCS) | FCS_EAP_EXT EAP-TLS<br>FCS_IPSEC_EXT IPsec |
| Identification and Authentication (FIA) | FIA_BMA_EXT Biometric Activation<br>FIA_PSK_EXT Pre-Shared Key Composition |
| Packet Filtering (FPF) | FPF_MFA_EXT Multifactor Authentication Filtering |
| Protection of the TSF (FPT) | FPT_TST_EXT TSF Self-Test |

## 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_IPSEC_EXT IPsec

#### Family Behavior

Components in this family describe requirements for IPsec implementation.

#### Component Leveling

FCS_IPSEC_EXT — 1

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 [**assignment**: *supported cryptographic 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 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 [**assignment**: *supported cryptographic algorithms* ] .

#### 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 [**selection**:*
  - *can be configured by [**selection**: an administrator, a VPN gateway] based on [**selection**: number of packets/number of bytes, length of time]*
  - *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 eight hours or less for Phase 2 SAs.

#### FCS_IPSEC_EXT.1.8

The TSF shall ensure that IKE protocols implement DH Groups

- 20 (384-bit Random ECP) according to RFC 5114 and

[**selection**:

- *15 (3072-bit MODP)*
- *16 (4096-bit MODP)*
- *17 (6144-bit MODP)*
- *18 (8192-bit MODP)*

] according to RFC 3526.

#### 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.1 (or FCS_RBG_EXT.1 in the case of , 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 [**selection**:

- *IKEv1 performs peer authentication using [**selection**: RSA, ECDSA] that use X.509v3 certificates* *that conform to RFC 4945.*

- *IKEv2 performs peer authentication using [**selection**: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [**selection**: Pre-shared Keys that conform to RFC 8784, Pre-shared Keys transmitted via EAP-TTLS, EAP-TLS, 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.2.1.2 FCS_EAP_EXT EAP-TLS

### Family Behavior

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

### Component Leveling

FCS_EAP_EXT ——— 1

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 support [**selection**: *EAP-TLS as specified in RFC 5216 and updated by RFC 8996* , *EAP-TTLS as specified in RFC 5281 and updated by RFC 8996* ] over a protected channel per the Base-PP with an authentication server.

#### FCS_EAP_EXT.1.2

The TSF shall implement [**selection**: *EAP-TLS*, *EAP-TTLS*] with the TSF as the EAP client, an external authentication server as the EAP server and the VPN peer as the supplicant.

#### FCS_EAP_EXT.1.3

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)

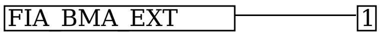
This PP-Module defines the following extended components as part of the FIA class originally defined by CC Part 2:

## C.2.2.1 FIA_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

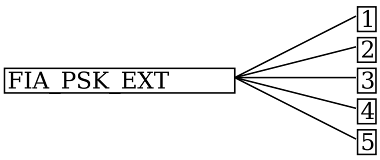
The TSF shall leverage the platform biometric features to confirm the user before initiating a trusted channel.

### C.2.2.2 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 [**selection**: *IKEv2*, *multifactor authentication filtering*].

##### 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 Pre-Shared Key Composition |

##### FIA_PSK_EXT.2.1

The TSF shall be able to [**selection**: *accept externally generated pre-shared keys, generate 256 bit-based pre-shared keys via the random number generator used by the TSF.*].

#### 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: | FCS_RBG.1 Random Bit Generation (RBG) FIA_PSK_EXT.1 Pre-Shared Key Composition |

##### 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 [**assignment**: *cryptographic algorithm used for key derivation* ] , with [**assignment**: *positive integer of 4096 or more* ] iterations, and output cryptographic key sizes [**assignment**: *output key size* ] that meet the following: [**assignment**: *list of standards* ] .

##### 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.1 and with entropy of [**assignment**: *value equal to or greater than 256* ] 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 Pre-Shared Key Composition

#### FIA_PSK_EXT.4.1

The TSF shall accept and send an HOTP while initiating a VPN 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 Pre-Shared Key Composition

#### FIA_PSK_EXT.5.1

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

## C.2.3 Packet Filtering (FPF)

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

### 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

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)

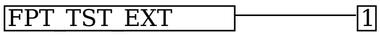
This PP-Module defines the following extended components as part of the FPT class originally defined by CC Part 2:

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

#### Management: FPT_TST_EXT.1

No specific management functions are identified.

#### Audit: FPT_TST_EXT.1

There are no auditable events foreseen.

#### FPT_TST_EXT.1 TSF Self-Test

Hierarchical to: No other components.

Dependencies to: No dependencies.

##### FPT_TST_EXT.1.1

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

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*].

# Appendix D - Acronyms

**Table 4: 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 (MD) |
| NAT | Network Address Translation |
| NIST | National Institute of Standards and Technology |
| OCSP | Online Certificate Status Protocol |
| OE | Operational Environment |
| OS | Operating System (OS) |
| 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 E - Bibliography

**Table 5: Bibliography**


| Identifier | Title |
| --- | --- |
| [CC] | Common Criteria for Information Technology Security Evaluation - <br>• Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022.<br>• Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022.<br>• Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022.<br>• Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022.<br>• Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022, Revision 1, November 2022. |
| [CEM] | Common Methodology for Information Technology Security Evaluation - <br>• Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022. |