# PP-Module for File Encryption Enterprise Management


Version: 1.0

2019-07-30

**National Information Assurance Partnership**

## Revision History

| Version | Date | Comment |
|---|---|---|
| 1.0 | 2019-07-30 | Initial Release |

## Contents

# 1 Introduction

## 1.1 Overview

The scope of the File Encryption Enterprise Management PP-Module is to describe the security functionality of a file encryption enterprise management product 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-PP

- Application Software Protection Profile, Version 1.3

This Base-PP is valid because a file encryption enterprise management product is a 3rd party application. The use case for a product conforming to the FE module is to protect data at rest on a device that is lost or stolen while powered off without any prior access by an adversary. The use case where an adversary obtains a device that is in a powered state and is able to make modifications to the environment or the TOE itself (e.g., evil maid attacks) is not addressed by that module. The module does contain protections to mitigate the potential for attack with a powered on device, but they are not sufficient to protect data from a skilled adversary with physical access.

While that use case is still true for the Enterprise Management PP-Module, this PP-module also expands the use case to include protecting the communications between the Enterprise Management Server and the client device through the use of a trusted channel. It also expands the use case to include the optional abilities of the EM to interact with clients (with proper authorization), to direct it to perform sanitation of keys and material on the device, to manage and store parts of the key chain required for decryption on the client, or to issue a recovery credential to reset the authentication factor if it has been lost.

The TOE and Its Supporting Environment:
The environment in which the EM functions is expected to exist is on a back end server, not on the endpoint system. It is expected to have secure access to a management system (e.g. Active Directory) and access to a means of storing key material when not in use. The EM shall not have the ability to access the secured stored key material without verification of access authority by the LDAP.

The Operating System environment may make a full range of services available to the Enterprise Management PP-Module, including hardware drivers, cryptographic libraries, and perhaps other services external to the TOE. The EM TOE may include or leverage features and functions within the operational environment.

## 1.2 Terms

The following sections provide both Common Criteria and technology terms used in this PP-Module.

### 1.2.1 Common Criteria Terms

| | |
|---|---|
| Assurance | Grounds for confidence that a TOE meets the SFRs [CC1]. |
| Common Criteria (CC) | Common Criteria for Information Technology Security Evaluation. |
| Distributed TOE | A TOE composed of multiple components operating as a logical whole. Specifically for the FE EM, it is an FE EM solution with multiple FE endpoints. |
| Operational Environment | Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy, including the platform, its firmware, and the operating system. |
| 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 (PP-Module) | An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles. |
| Security Target (ST) | A set of implementation-dependent security requirements for a specific product. |
| Target of Evaluation (TOE) | The product under evaluation. In this case, file encryption enterprise management software and its supporting documentation. |
| TOE Security Functionality (TSF) | The security functionality of the product under evaluation. |
| TOE Summary Specification (TSS) | A description of how a TOE satisfies the SFRs in a ST. |
| Security Functional Requirement (SFR) | A requirement for security enforcement by the TOE. |
| Security Assurance Requirement (SAR) | A requirement to assure the security of the TOE. |

### 1.2.2 Technical Terms

| | |
|---|---|
| Authorization factor (AF) | A value that a user knows, has, or is (e.g. password, token, etc.) submitted to the TOE to establish that the user is in the community authorized to access the requested material. |
| Entropy | This cryptographic function provides a seed for a random bit generator by accumulating the |

| | |
|---|---|
| Source | 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. |
| Key Sanitization | A method of sanitizing encrypted data by securely overwriting the key, as described in the key destruction requirement, that was encrypting the data. |
| File/Set of files | The user data that is selected to be encrypted, which can include individual file encryption (with a FEK per file) or a set of files encrypted with a single FEK. |
| File Encryption Key (FEK) | The key that is used by the encryption algorithm to encrypt the selected user data on the host machine. |
| Key Chaining | The method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key which encrypts the data; this method can have any number of layers. |
| Key Encryption Key (KEK) | The key that is used to encrypt another key. |
| Keying Escrow | The process of exporting a key to an alternate location. |
| Keying material | Key material is commonly known as critical security parameter (CSP) data, and also includes authorization data, nonces, and metadata. |
| Key Release Key | A key used to release another key from storage, it is not used for the direct derivation or decryption of another key. |
| Noise Source | The component of an RBG that contains the non-deterministic, entropy-producing activity. |
| Non-Volatile Memory | A type of computer memory that will retain information without power. |
| Powered-Off State | The device has been shut down. |
| Protected Data | This refers to all files designated by the user for encryption. |
| Random Bit Generator (RBG) | A cryptographic function composed of an entropy source and DRBG that is invoked for random bits needed to produce keying material. |
| Registration | The initial process of associating and endpoint and/or user with the server. |
| Submask | A submask is a bit string that can be generated and stored in a number of ways. |
| System Identity | A composition of a series of identifiers that may vary, but aim to identity and associate with a specific system. |

## 1.3 Compliant Targets of Evaluation

The target of evaluation for this PP-Module is the Enterprise Management (EM) function of a FE solution. The following section provides an overview of the security functionality of this PP-module.

### 1.3.1 TOE Boundary

The application, which consists of the software provided by its vendor, is installed onto the platform(s) it operates on. It executes on the platform, which may be an operating system, hardware environment, a software based execution environment, or some combination of these. Those platforms may themselves run within other environments, such as virtual machines or operating systems, that completely abstract away the underlying hardware from the application. The TOE is not accountable for security functionality that is implemented by platform layers that are abstracted away. Some evaluation activities are specific to the particular platform on which the application runs, in order to provide precision and repeatability. The only platforms currently recognized by [AppPP] and this module are those specified in SFR Evaluation Activities. To test on a platform for which there are no EAs, a Vendor should contact NIAP with recommended EAs. NIAP will determine if the proposed platform is appropriate for the PP and accept, reject, or develop EAs as necessary in coordination with the technical community.

The TOE includes any software in the application installation package, even those pieces that may extend or modify the functionality of the underlying platform, such as kernel drivers. BIOS and other firmware, the operating system kernel, and other systems software (and drivers) provided as part of the platform are outside the scope of this document.

## 1.4 Use Cases

**[USE CASE 1] Enterprise Management**

The use case for this PP-Module is protecting the communications between the Enterprise Management Server and the client device through the use of a trusted channel. Including the optional abilities of the EM to interact with clients (with proper authorization), to direct it to perform sanitation of keys and material on the device, to manage and store parts of the key chain required for decryption on the client, or to issue a recovery credential to reset the authentication factor if it has been lost.

# 2 Conformance Claims

### Conformance Statement

This PP-Module inherits exact conformance as required from the specified Base-PP and as defined in the [CC] and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017). This PP-Module is conformant to Parts 2 (extended) and 3 (extended) of Common Criteria Version 3.1, Revision 5 [CC]. The following PPs and PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module.

- PP-Module for VPN Client, Version 2.1
- PP-Module for File Encryption, Version 2.0

If claiming compliance to a PP-Configuration that includes multiple PP-Modules, the ST author must ensure any duplicative SFRs are iterated using unique identifiers. This will allow the reader to easily determine which iteration applies to each TOE component.

### Package Claims

This PP-Module is TLS Package Version 1.1 Conformant.

# 3 Security Problem Description

The primary asset that is being protected is the sensitive user data stored on a system. The threat model thus focuses on a host machine that has been compromised by an unauthorized user. This section addresses threats to the TOE only.

## 3.1 Threats

**T.KEYING_MATERIAL_COMPROMISE_SERVER**
Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. This PP-Module considers possession of key material of equal importance to the data itself. Threat agents may look for key material in unencrypted storage on the Management Server and in external databases in the operating environment (OE), e.g. SQL database.

**T.MAN_IN_THE_MIDDLE**
An attacker listening on the communication between the Management Server and the Client(s) to obtain the user's credential, keys, or recovery material.

**T.UNAUTHORIZED_ADMINISTRATOR_ACCESS**
An attacker masquerading as an administrator to the Management Server to gain access to TOE management functionality to gain unauthorized access to protected data or prevent legitimate users from gaining authorized access.

**T.UNTRUSTED_COMMUNICATION_CHANNELS**
An attacker targeting the Management Server using insecure tunneling protocols or the presence of an unencrypted path to disclose keys, key material, or recovery material transferred between the endpoint and the Management Server.

**T.UNAUTHORIZED_DATA_ACCESS_ENDPOINT**
An attacker accessing the data on the encrypted file(s) by getting access to a protected file(s), attaching it to a host system controlled by the attacker and using the key material, or optionally a recovery credential to access the data. The file encryption module addresses the primary threat of unauthorized disclosure of recovery material.

**T.UNAUTHORIZED_DATA_ACCESS_SERVER**
An attacker accessing the Management Server and generating a recovery key chain for an endpoint. The File Encryption PP-Module addresses the primary threat of unauthorized disclosure of data protected on the endpoint; this adds the Management Server to the scope of the threat.

## 3.2 Assumptions

These assumptions are made on the Operational Environment in order to be able to ensure that the security functionality specified in the PP-Module can be provided by the TOE. If the TOE is placed in an Operational Environment that does not meet these assumptions, the TOE may no longer be able to provide all of its security functionality.

**A.ENVIRONMENTAL_STORAGE**
Any key storage mechanism provided by the Operational Environment is able to provide the same level of security as a TOE-internal storage mechanism that is conformant to this PP-Configuration.

**A.PHYSICAL_SERVER**
The platform on which the Management Server resides is physically protected in its Operational Environment and not subject to physical attacks that compromise the security and/or interfere with the platform's correct operation.

**A.SECURED_CONFIGURATION**
The Management Server and the remote endpoints are installed and configured in accordance with their evaluated configuration.

**A.SECURED_ENVIRONMENT**
Any environmental components required to support the functionality of the Management Server (e.g. underlying operating system, firewall, database) are installed and configured in accordance with its proper configuration.

## 3.3 Organizational Security Policies

There are no Organizational Security Policies for the PP-Module.

# 4 Security Objectives

The Security Problem described in Section 3 will be addressed by a combination of cryptographic capabilities. Compliant TOEs will provide security functionality that addresses threats to the TOE and enforces policies that are imposed by law and regulation. The following subsections provide a description of the security objectives required to meet the threats/policies previously discussed. The description of these security objectives are in addition to that described in the [AppPP].

Note: in each subsection below particular security objectives are identified (highlighted by O.) and they are matched with the associated security functional requirements (SFRs) that provide the mechanisms to satisfy the objectives.

The Security Objectives are the requirements for the Target of Evaluation (TOE) and for the Operational Environment derived from the threats in Section 3.

## 4.1 Security Objectives for the TOE

**O.ENTERPRISE_KEY_PROTECTION**

**Protection of Key Material:** FPT_KYP_EXT.2 requires that the key material, and optionally recovery credentials be uniquely associated with the endpoint at a minimum. Additionally, key material may also be associated with a specific system or user to prevent an attacker from accessing the data on the endpoint by transferring the data in a host with weaker security. A product which distributes keys to meet the requirements of FPT_KYP_EXT.2 will additionally prevent an attacker from gaining access to the encrypted data.

Addressed by: FPT_KYP_EXT.2

**O.KEY_MATERIAL_SERVER**

**Key Material Server:** The keying material that threat agents may attempt to compromise are generated by the TOE as specified by FCS_CKM.1(2). One or more submasks may be utilized on the endpoint to protect the FEK or a KEK, part of the keychain to protect that is stored on the server for additional authorization or recovery. The server key chain can be maintained by several methods, including:

- Key establishment [FCS_CKM.2]
- Key derivation [FCS_KDF_EXT.1]
- Key wrapping [FCS_COP.1(5)]
- Key combining [FCS_SMC_EXT.1]
- Key encryption [FCS_COP.1(7)]
- Key Transport [FCS_COP.1(6)]

Key chains may be maintained using asymmetric [FCS_CKM_EXT.1] and/or symmetric [FCS_CKM.1(2)].

These requirements ensure keys are properly generated and protected. If selected, FMT_MOF.1 ensures that only administrators can select the encryption algorithms and key sizes. Only administrators can perform management functions on the Enterprise Management Server as defined in FMT_SMF.1.

FCS_KYC_EXT.1 extends the requirements of File Encryption PP-Module key chaining to key chains generated or maintained by the Server.

FPT_ITT.1 ensures that keys and key material transported between the EM and the endpoint are protected.

FPT_KYP_EXT.1 ensures unwrapped key material is not stored in non-volatile memory minimizing the exposure of plaintext keys and key material.

FTP_DIT_EXT.1 specifies the protocols used to ensure that key material is not exposed through the communication channel between an Enterprise Server and the endpoint. The requirements for establishing keys are validated by FCS_CKM.2 which relies on the SFRs or package claims selected in FTP_DIT_EXT.1 to implement secure communications. The various iterations of FCS_COP.1 as well as FCS_RBG_EXT.1 all validate that the cryptography used to initiate and protect the communication channel protocols between the Enterprise Server and the endpoint, if remote management is supported by the TSF. If implemented on the server, FCS_CKM_EXT.4 ensures proper destruction of keys and key material on the server when no longer needed.

In order to ensure that a key is only released to the appropriate endpoint, FCS_KYP_EXT.3 ensures that there is attribution of the endpoint or encrypted file(s) and a key. The optional Server requirement FCS_CKM.2 ensures that if a key is communicated between the server and the endpoint, keys distributed by the server are given to the correct endpoint for the purpose of delivering a key.

To ensure the key chain is started properly, FIA_AUT_EXT.1 defines proper authentication and conditioning.

Addressed by: FCS_CKM.1(2) (from Base-PP), FCS_CKM_EXT.1 (from Base-PP), FCS_CKM.2 (from Base-PP), FTP_DIT_EXT.1 (modified from Base-PP), FCS_CKM_EXT.4, FCS_COP.1(5), FCS_COP.1(6), FCS_COP.1(7), FCS_KDF_EXT.1, FCS_KYC_EXT.1, FCS_KYP_EXT.3, FCS_SMC_EXT.1, FIA_AUT_EXT.1, FMT_MOF.1, FMT_SMF.1, FPT_ITT.1

**O.RECOVERY_PROTECTION**

**Recovery Protection:** FIA_UAU.1 requires the administrator to be authenticated prior to allowing the administrator to manage the product via the remote console. FIA_UID.1 requires the admin to be identified prior to allowing the administrator to manage the product via the remote console. FMT_MTD.1 requires that actions which result in changes to key material, user authentication policy and recovery are constrained to administrators and specific times. FMT_SMR.2 requires users be assigned roles. FCS_VAL_EXT.1(2) requires user authentication to be validated by the Operational Environment or the TOE prior to releasing a recovery value and FCS_VAL_EXT.2(2) specifies what happens if the validation fails. Recovery methods are defined by FIA_REC_EXT.1 and FIA_CHR_EXT.1.

The optional capability which may be provided by the TSF would include encryption of data stored on the server, as validated by FCS_COP.1(1); and certificate-based authentication, validated by FIA_X509_EXT.2 and validation, as validated by FIA_X509_EXT.1.

Addressed by: FCS_COP.1(1) (from Base-PP), FIA_X509_EXT.1 (from Base-PP), FIA_X509_EXT.2 (from Base-PP), FCS_VAL_EXT.1(2), FCS_VAL_EXT.2(2), FIA_REC_EXT.1, FIA_UAU.1, FIA_UID.1, FMT_MTD.1, FMT_SMR.2, FIA_CHR_EXT.1 (selection-based)

**O.SECURE_CHANNEL**

**Secure Channel:** FPT_ITT.1 ensures protection of intra TOE communication and FTP_DIT_EXT.1 covers all transmitted sensitive data. If server side key generation is implemented, FCS_CKM.1(1) ensures sufficiently strong keys correctly generated on the server to meet the requirements of FTP_TRP.1. Products implementing cryptographic communication protocols between the server and managed endpoints must meet the requirements for the specific protocols as defined in any of {FCS_HTTPS_EXT.1, PP-Module for VPN Client, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1}

If the EM Server generates signatures to request or verify certificates, FCS_COP.1(1) ensures correct cryptographic operation in signature generation process.

The TOE is not required to support remote administration. If it does, FTP_TRP.1 addresses the threat of disclosure of keys, key material, or recovery material transferred between the endpoint or a remote administrator and the Management Server when transmitted over untrusted communication channels by requiring use of IPsec, SSH, TLS, and/or TLS/HTTPS protocols when such data passes through those channels.

FTP_DIT_EXT.1, which is modified from its definition in the Base-PP to mandate the use of at least one trusted communications protocol, specifies the protocols used to ensure that data in transit over this channel is secured.

FIA_X509_EXT.1, FIA_X509_EXT.2, and FIA_X509_EXT.3 ensure the communication channel is established only with a server that is authenticated. FCS_COP.1(1) ensures correct generation of cryptographic signatures.

If the TSF generates password authorization factors, the requirements of FCS_PCC_EXT.1 and FCS_CKM_EXT.6 ensure that the password data is not subjected to unauthorized disclosure or brute force attack.

Addressed by: FTP_DIT_EXT.1 (modified from Base-PP), FCS_CKM.1(1) (from Base-PP), FCS_COP.1(1) (from Base-PP), FCS_COP.1(3) (from Base-PP), FCS_RBG_EXT.1 (from Base-PP), FIA_X509_EXT.1 (from Base-PP), FIA_X509_EXT.2 (from Base-PP), FIA_X509_EXT.3 (from Base-PP), FPT_ITT.1, FCS_CKM_EXT.6 (selection-based), FTP_TRP.1 (selection-based)

**O.VERIFIED_ADMIN**

**Verified Admin:** FIA_UAU.1 requires that the administrator be authenticated by the EM, which is verified by FCS_VAL_EXT.1(1). If the TSF is responsible for this verification, FCS_VAL_EXT.2(1) describes the behavior that the TOE enforces if the validation fails. The administrator is required by FIA_UID.1 to successfully authenticate to the EM prior to being permitted to perform management functions.

Addressed by: FCS_VAL_EXT.1(1), FIA_UAU.1, FIA_UID.1, FCS_VAL_EXT.2(1) (selection-based)

## 4.2 Security Objectives for the Operational Environment

The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). The security objectives for the Operational Environment consist of a set of statements describing the goals that the Operational Environment should achieve. This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 3 are incorporated as security objectives for the environment.

**OE.ENVIRONMENTAL_STORAGE**

If the TOE relies on the Operational Environment for key storage, the storage mechanism will provide at least the same level of security as a TOE-internal storage mechanism as defined by FPT_KYP_EXT.1.

**OE.PHYSICAL_SERVER**

The Operating environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE as defined in FCS_KYC_EXT.1.

**OE.SECURED_CONFIGURATION**

The Management Server and remote endpoints are configured in accordance with its associated operational guidance so that the level of security that is provided by the TOE is consistent with its evaluated configuration.

**OE.SECURED_ENVIRONMENT**

The components of the Management Server’s underlying platform are configured in accordance with their associated operational guidance so that the TOE is deployed in an environment that is consistent with its evaluated configuration.

## 4.3 Security Objectives Rationale

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

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.KEYING_MATERIAL_COMPROMISE_SERVER | O.KEY_MATERIAL_SERVER | The threat T.KEYING_MATERIAL_COMPROMISE is countered by O.KEY_MATERIAL_SE this provides for proper protection of k material on the server. |
| T.MAN_IN_THE_MIDDLE | O.SECURE_CHANNEL | The threat T.MAN_IN_THE_MIDDLE i countered by O.SECURE_CHANNEL a protects against man in the middles at |

| | | |
|---|---|---|
| T.UNAUTHORIZED_ADMINISTRATOR_ACCESS | O.VERIFIED_ADMIN | The threat T.UNAUTHORIZED_ADMINISTRATOR is countered by O.VERIFIED_ADMIN a provides methods to verify the adminis |
| T.UNTRUSTED_COMMUNICATION_CHANNELS | O.SECURE_CHANNEL | The threat T.UNTRUSTED_COMMUNICATION_C is countered by O.SECURE_CHANNEL provides a trusted channel for any ser endpoint communications. |
| T.UNAUTHORIZED_DATA_ACCESS_ENDPOINT | O.ENTERPRISE_KEY_PROTECTION | The threat T.UNAUTHORIZED_DATA_ACCESS_E is countered by O.ENTERPRISE_KEY_PROTECTION as provides for encryption of keys that pr |
| T.UNAUTHORIZED_DATA_ACCESS_SERVER | O.RECOVERY_PROTECTION | The threat T.UNAUTHORIZED_DATA_ACCESS_S countered by O.RECOVERY_PROTECT this provides for protection of recovery information. |
| A.ENVIRONMENTAL_STORAGE | OE.ENVIRONMENTAL_STORAGE | The operational environment objective OE.ENVIRONMENTAL_STORAGE is r through A.ENVIRONMENTAL_STORA |
| A.PHYSICAL_SERVER | OE.PHYSICAL_SERVER | The operational environment objective OE.PHYSICAL_SERVER is realized thr A.PHYSICAL_SERVER. |
| A.SECURED_CONFIGURATION | OE.SECURED_CONFIGURATION | The operational environment objective OE.SECURED_CONFIGURATION is re through A.SECURED_CONFIGURATIO |
| A.SECURED_ENVIRONMENT | OE.SECURED_ENVIRONMENT | The operational environment objective OE.SECURED_ENVIRONMENT is real through A.SECURED_ENVIRONMENT |

# 5 Security Requirements

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

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: are identified with a number inside parentheses (e.g. "(1)")

## 5.1 App PP Security Functional Requirements Direction

The TOE is expected to rely on some of the security functions implemented by the application as a whole and evaluated against [AppPP]. The following section 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 5.2.

### 5.1.1 Modified SFRs

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

#### 5.1.1.1 Trusted Path/Channel (FTP)

**FTP_DIT_EXT.1 Protection of Data in Transit**

FTP_DIT_EXT.1.1

The TSF shall [**selection**:

- *encrypt all transmitted [**selection**: sensitive data, data] with [**selection**: HTTPS in accordance with FCS_HTTPS_EXT.1 **(from [AppPP])**, TLS as defined in the TLS Package, DTLS as defined in the TLS Package, SSH as conforming to the Extended Package for Secure Shell] ,*
- *invoke platform-provided functionality to encrypt all transmitted sensitive data with [**selection**: HTTPS, TLS, DTLS, SSH] ,*
- *invoke platform-provided functionality to encrypt all transmitted data with [**selection**: HTTPS, TLS, DTLS, SSH]*

] between itself and another trusted IT product.

**Application Note:** This SFR is modified from its definition in the Base-PP by removing the first selection (where the application does not transmit any data or sensitive data). By definition, a TOE that conforms to this PP-Module must have the ability to transmit sensitive data to another trusted IT product.
If *encrypt all transmitted* is selected and *TLS* is selected, then evaluation of elements from either FCS_TLSC_EXT.1 or FCS_TLSS_EXT.1 is required.

If *encrypt all transmitted* is selected and *HTTPS* is selected, FCS_HTTPS_EXT.1 is required.

If *encrypt all transmitted* is selected and *DTLS* is selected, FCS_DTLS_EXT.1 is required.

If *encrypt all transmitted* is selected and *SSH* is selected, the TSF is required to be validated against the *Extended Package for Secure Shell*.

If *encrypt all transmitted* is selected the corresponding FCS_COP.1 requirements will be included.

**Evaluation Activity**

*The evaluator shall perform the activities below in addition to the evaluation activities required for this SFR in the Base-PP.*
***TSS***
*For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality.*

***Guidance***
*None.*

***Tests***
*The evaluator shall perform the following tests.*

- ***Test 1:** The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, or SSH in accordance with the selection in the ST.*
- ***Test 2:** The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.*

- ***Test 3:*** *The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found.*
  ***For iOS:****If the platformIf "encrypt all transmitted data" is selected, the evaluator shall ensure that the application's Info.plist file does not contain the NSAllowsArbitraryLoads or NSExceptionAllowsInsecureHTTPLoads keys, as these keys disable iOS's Application Transport Security feature.*

## 5.2 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.

### 5.2.1 Cryptographic Support (FCS)

#### FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**selection**:

- *For volatile memory, the destruction shall be executed by a [**selection**:*
  - *single overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's RBG, zeroes, ones, new value of a key, [**assignment**: any value that does not contain any CSP]] ,*
  - *removal of power to the memory,*
  - *destruction of reference to the key directly followed by a request for garbage collection*

  *] ,*
- *For non-volatile memory, the destruction shall be executed by [**selection**:*
  - *destruction of all KEKs protecting the target key, where none of the KEKs protecting the target key are derived ,*
  - *the invocation of an interface provided by the underlying platform that [**selection**:*
    - *logically addresses the storage location of the key and performs a [**selection**: single, [**assignment**: ST author defined multi-pass]] overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's RBG, zeroes, ones, new value of a key, [**assignment**: any value that does not contain any CSP]] ,*
    - *instructs the underlying platform to destroy the abstraction that represents the key*

    *]*

  *]*

].

**Application Note:** The interface referenced in the requirement could take different forms, the most likely of which is an application programming interface to an OS kernel. There may be various levels of abstraction visible. For instance, in a given implementation that overwrites a key stored in non-volatile memory, the application may have access to the file system details and may be able to logically address specific memory locations. In another implementation that instructs the underlying platform to destroy the representation of a key stored in non-volatile memory, the application may simply have a handle to a resource and can only ask the platform to delete the resource, as may be the case with a platforms secure key store. The latter implementation should only be used for the most restricted access. The level of detail to which the TOE has access will be reflected in the TSS section of the ST. Several selections allow assignment of a 'value that does not contain any CSP'. This means that the TOE uses some other specified data not drawn from a source that may contain key material or reveal information about key material, and not being any of the particular values listed as other selection options. The point of the phrase 'does not contain any CSP' is to ensure that the overwritten data is carefully selected, and not taken from a general 'pool' that might contain current or residual data that itself requires confidentiality protection.

For the selection "destruction of all KEKs protecting target key, where none of the KEKs protecting the target key are derived", a key can be considered destroyed by destroying the key that protects the key. If a key is wrapped or encrypted it is not necessary to "overwrite" that key, overwriting the key that is used to wrap or encrypt the key used to encrypt/decrypt data, using the appropriate method for the memory type involved, will suffice. For example, if a product uses a Key Encryption Key (KEK) to encrypt a File Encryption Key (FEK), destroying the KEK using one of the methods in FCS_CKM_EXT.4 is sufficient, since the FEK would no longer be usable (of course, presumes the FEK is still encrypted and the KEK cannot be recovered or re-derived).

FCS_CKM_EXT.4.2

The TSF shall destroy all keys and key material when no longer needed.

**Application Note:** Keys, including intermediate keys and key material that are

no longer needed are destroyed by using an approved method, FCS_CKM_EXT.4.1. Examples of keys are intermediate keys, submasks. There may be instances where keys or key material that are contained in persistent storage are no longer needed and require destruction. Base on their implementation, vendors will explain when certain keys are no longer needed. There are multiple situations in which key material is no longer necessary, for example, a wrapped key may need to be destroyed when a password is changed. However, there are instances when keys are allowed to remain in memory, for example, a device identification key. If a PIN was used for a smart card and managed by the TOE, ensuring that the PIN was properly destroyed must be addressed.

**Evaluation Activity**

***TSS***
*The evaluator shall verify the TSS provides a high level description of what it means for keys and key material to be no longer needed and when they should be expected to be destroyed. The evaluator shall verify the TSS provides a high level description of what it means for keys and key material to be no longer needed and when then should be expected to be destroyed.*

***KMD***

*The evaluator examines the KMD to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in non-volatile memory) and how they are overwritten.*

*The evaluator shall check to ensure the KMD lists each type of key that is stored in in non-volatile memory, and identifies how the TOE interacts with the underlying platform to manage keys (e.g., store, retrieve, destroy). The description includes details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys (e.g., file system APIs, platform key store APIs).*

*The evaluator examines the interface description for each different media type to ensure that the interface supports the selection(s) and description in the KMD.*

*If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the KMD to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.*

*The evaluator shall check that the KMD identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.*

*If the selection "destruction of all KEKs protecting target key, where none of the KEKs protecting the target key are derived" is included the evaluator shall examine the TOE's keychain in the KMD and identify each instance when a key is destroyed by this method. In each instance the evaluator shall verify all keys capable of decrypting the target key are destroyed in accordance with a specified key destruction method in FCS_CKM_EXT.4.1. The evaluator shall verify that all of the keys capable of decrypting the target key are not able to be derived to reestablish the keychain after their destruction.*

*The evaluator shall verify the KMD includes a description of the areas where keys and key material reside and when the keys and key material are no longer needed.*

*The evaluator shall verify the KMD includes a key lifecycle, that includes a description where key material reside, how the key material is used, how it is determined that keys and key material are no longer needed, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM_EXT.4.1 for the destruction.*

***Guidance***
*There are a variety of concerns that may prevent or delay key destruction in some cases.*

*The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information.*

*The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer and how such situations can be avoided or mitigated if possible.*

*Some examples of what is expected to be in the documentation are provided here.*

*When the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage*

*collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, to mitigate this the drive should support the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.*

*Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. To reduce this risk, the operating system and file system of the OE should support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion. If a RAID array is being used, only set-ups that support TRIM are utilized. If the drive is connected via PCI-Express, the operating system supports TRIM over that channel.*

*The drive should be healthy and contains minimal corrupted data and should be end of lifed before a significant amount of damage to drive health occurs, this minimizes the risk that small amounts of potentially recoverable data may remain in damaged areas of the drive.*

***Tests***
*These tests are only for key destruction provided by the application, test 2 does not apply to any keys using the selection "new value of a key":*

- ***Test 1:*** *Applied to each key held in volatile memory and subject to destruction by overwrite by the TOE (whether or not the value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the key destruction method was removal of power, then this test is unnecessary.*

  *The evaluator shall:*

  *1. Record the value of the key in the TOE subject to clearing.*
  *2. Cause the cause the TOE or the underlying platform to dump to perform a normal cryptographic processing with the key from Step #1.*
  *3. Cause the TOE to clear the key.*
  *4. Cause the TOE to stop the execution but not exit.*
  *5. Cause the TOE to dump the entire memory of the TOE into a binary file.*
  *6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.*
  *Steps #1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.*

- ***Test 2:*** *[Conditional] If new value of a key is selected this test does not apply.*
  *Applied to each key held in non-volatile memory and subject to destruction by the TOE.*
  *The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to ensure the tests function as intended.*
  *1. Identify the purpose of the key and what access should fail when it is deleted. (e.g. the file encryption key being deleted would cause data decryption to fail.)*
  *2. Cause the TOE to clear the key.*
  *3. Have the TOE attempt the functionality that the cleared key would be necessary for.*
  *4. The test succeeds if Step #3 fails.*

  *Tests 3 and 4 do not apply for the selection instructing the underlying platform to destroy the representation of the key, as the TOE has no visibility into the inner workings and completely relies on the underlying platform.*

- ***Test 3:*** *Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media (e.g., MBR file system):*
  *1. Record the value of the key in the TOE subject to clearing.*
  *2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
  *3. Cause the TOE to clear the key.*
  *4. Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.*

- ***Test 4:*** *Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media:*
  *1. Record the logical storage location of the key in the TOE subject to clearing.*
  *2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
  *3. Cause the TOE to clear the key.*
  *4. Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.*
  *The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.*

**FCS_COP.1(5) Cryptographic operation (Key Wrapping)**

FCS_COP.1.1(5)

The TSF shall **[selection:**

- ***not perform key wrapping,***
- ***use platform-provided functionality to perform Key Wrapping,***
- ***implement functionality to perform Key Wrapping in accordance with a specified cryptographic algorithm [AES] in the following modes [selection:***
  - ***Key Wrap,***
  - ***Key Wrap with Padding,***
  - ***GCM mode,***
  - ***CCM mode***

  ***] and the cryptographic key size [selection: 128 bits (AES), 256 bits (AES)] that meet the following: [selection:***
  - ***"NIST SP 800-38C",***
  - ***"NIST SP 800-38D",***
  - ***"NIST SP 800-38F"***

  ***] and no other standards***

**]**.

**Application Note:** This applies to any key wrapping occurring on the enterprise server. This requirement is used in the body of the ST if the ST author chooses to use key wrapping in the key chaining approach that is specified in FCS_KYC_EXT.1.

**Evaluation Activity**

***TSS***
*Conditional: If not perform key wrapping was selected, then the evaluator shall only examine the TSS to verify no key wrapping is performed.*
*Conditional: If use platform provided functionality was selected, then the evaluator shall examine the TSS to verify that it describes how the FEK encryption/decryption is invoked.*
*Conditional: If implement functionality was selected, the evaluator shall check that the TSS includes a description of encryption function(s) used for key wrapping. The evaluator should check that this description of the selected encryption function includes the key sizes and modes of operations as specified in the selections above. The evaluator shall check that the TSS describes the means by which the TOE satisfies constraints on algorithm parameters included in the selections made for 'cryptographic algorithm' and 'list of standards'.*

*The evaluator shall verify the TSS includes a description of the key wrap function(s) and shall verify the key wrap uses an approved key wrap algorithm according to the appropriate specification.*

***KMD***

*The evaluator shall review the KMD to ensure that all keys are wrapped using the approved method and a description of when the key wrapping occurs.*

***Guidance***
*If multiple encryption modes are supported, the evaluator examines the guidance documentation to determine that the method of choosing a specific mode/key size is described.*

***Tests***
*Conditional: If 'not perform key wrapping' was selected, no testing is performed.*

*The assurance activity tests specified for AES in GCM mode in the underlying [AppPP] shall be performed in the case that "GCM" is selected in the requirement.*

***AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test***

*The evaluator will test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:*

- *128 and 256 bit key encryption keys (KEKs)*
- *Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).*

*using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator will use the AES-KW authenticated-encryption function of a known good implementation.*

*The evaluator will test the authenticated-decryption functionality of AES-KW*

*using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.*

*The evaluator will test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:*

- *One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).*
- *One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).*

*The evaluator will test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.*

*AES-CCM Tests*
*It is not recommended that evaluators use values obtained from static sources such as http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip or use values not generated expressly to exercise the AES-CCM implementation.*

*The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:*
*Keys: All supported and selected key sizes (e.g., 128, 256 bits).*

*Associated Data: Two or three values for associated data length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported associated data lengths, and 2^16 (65536) bytes, if supported.*
*Payload: Two values for payload length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported payload lengths.*

*Nonces: All supported nonce lengths (7, 8, 9, 10, 11, 12, 13) in bytes.*

*Tag: All supported tag lengths (4, 6, 8, 10, 12, 14, 16) in bytes.*

*The testing for CCM consists of five tests. To determine correctness in each of the below tests, the evaluator shall compare the ciphertext with the result of encryption of the same inputs with a known good implementation.*

*Variable Associated Data Test*
*For each supported key size and associated data length, and any supported payload length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.*

*Variable Payload Test*
*For each supported key size and payload length, and any supported associated data length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.*

*Variable Nonce Test*
*For each supported key size and nonce length, and any supported associated data length, payload length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.*

*Variable Tag Test*
*For each supported key size and tag length, and any supported associated data length, payload length, and nonce length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.*

*Decryption-Verification Process Test*
*To test the decryption-verification functionality of AES-CCM, for each combination of supported associated data length, payload length, nonce length, and tag length, the evaluator shall supply a key value and 15 sets of input plus ciphertext, and obtain the decrypted payload. Ten of the 15 input sets supplied should fail verification and five should pass.*

**FCS_COP.1(6) Cryptographic operation (Key Transport)**

FCS_COP.1.1(6)

The TSF shall **[selection:**

- ***not perform key transport,***
- ***perform [key transport] in accordance with a specified cryptographic algorithm [RSA in the following modes [selection: KTS-OAEP, KTS-KEM-KWS] and the cryptographic key size [selection: 3072, 4096]bits that meet the following: [NIST SP 800-56B, Revision 1].***

**]**.

**Application Note:** This requirement is used in the body of the ST if the ST author chooses to use key transport in the key chaining approach that is

specified in FCS_KYC_EXT.1.

**Evaluation Activity**

***TSS***
*Conditional: If 'not perform key transport' was selected, then the evaluator shall only examine the TSS to verify no key transport is performed.*
*The evaluator shall verify the TSS provides a high level description of the RSA scheme and the cryptographic key size that is being used, and that the asymmetric algorithm being used for key transport is RSA. If more than one scheme/key size are allowed, then the evaluator shall make sure and test all combinations of scheme and key size. There may be more than one key size to specify - an RSA modulus size (and/or encryption exponent size), an AES key size, hash sizes, MAC key/MAC tag size.*

*If the KTS-OAEP scheme was selected, the evaluator shall verify that the TSS identifies the hash function, the mask generating function, the random bit generator, the encryption primitive and decryption primitive. If the KTS-KEM-KWS scheme was selected, the evaluator shall verify that the TSS identifies the key derivation method, the AES-based key wrapping method, the secret value encapsulation technique, and the random number generator.*

***Guidance***
*None.*

***Tests***
*For each supported key transport schema, the evaluator shall initiate at least 25 sessions that require key transport with an independently developed remote instance of a key transport entity, using known RSA key-pairs. The evaluator shall observe traffic passed from the sender-side and to the receiver-side of the TOE, and shall perform the following tests, specific to which key transport scheme was employed. If the KTS-OAEP scheme was selected, the evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall inspect each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE and make sure it is the correct length, either 256 or 384 bytes depending on RSA key size. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts that are the wrong length and verify that the erroneous input is detected and that the decryption operation exits with an error code.*
- ***Test 2:*** *The evaluator shall convert each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE to the correct cipher text integer, c, and use the decryption primitive to compute em = RSADP((n,d),c) and convert em to the encoded message EM. The evaluator shall then check that the first byte of EM is 0x00. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts where the first byte of EM was set to a value other than 0x00, and verify that the erroneous input is detected and that the decryption operation exits with an error code.*
- ***Test 3:*** *The evaluator shall decrypt each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE using RSADP, and perform the OAEP decoding operation (described in NIST SP 800-56B section 7.2.2.4) to recover HA' \|\| X. For each HA', the evaluator shall take the corresponding A and the specified hash algorithm and verify that HA' = Hash(A). The evaluator shall also force the TOE to perform some RSA-OAEP decryption where the A value is passed incorrectly, and the evaluator shall verify that an error is detected.*
- ***Test 4:*** *The evaluator shall check the format of the 'X' string recovered in OAEP.Test.3 to ensure that the format is of the form PS \|\| 01 \|\| K, where PS consists of zero or more consecutive 0x00 bytes and K is the transported keying material. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts for which the resulting 'X' strings do not have the correct format (i.e., the leftmost non-zero byte is not 0x01). These incorrectly formatted 'X' variables shall be detected by the RSA-OEAP decrypt function.*
- ***Test 5:*** *The evaluator shall trigger all detectable decryption errors and validate that the returned error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations are revealed to the sender.*

*If the KTS-KEM-KWS scheme was selected, the evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall inspect each cipher text, C, produced by KTS-KEM-KWS encryption operation of the TOE and make sure the length (in bytes) of the cipher text, cLen, is greater than nLen (the length, in bytes, of the modulus of the RSA public key) and that cLen - nLen is consistent with the byte lengths supported by the key wrapping algorithm. The evaluator shall feed into the KTS-KEM-KWS decryption operation a cipher text of unsupported length and verify that an error is detected and that the decryption process stops.*
- ***Test 2:*** *The evaluator shall separate the cipher text, C, produced by the sender-side of the TOE into its C0 and C1 components and use the RSA decryption primitive to recover the secret value, Z, from C0. The evaluator shall check that the unsigned integer represented by Z is greater than 1 and less than n-1, where n is the modulus of the RSA*

*public key. The evaluator shall construct examples where the cipher text is created with a secret value Z = 1 and make sure the KTS-KEM-KWS decryption process detects the error. Similarly, the evaluator shall construct examples where the cipher text is created with a secret value Z = n - 1 and make sure the KTS-KEM-KWS decryption process detects the error.*

- ***Test 3:*** *The evaluator shall attempt to successfully recover the secret value Z, derive the key wrapping key, KWK, and unwrap the KWA-cipher text following the KTS-KEM-KWS decryption process given in NISP SP 800-56B section 7.2.3.4. If the key-wrapping algorithm is AES-CCM, the evaluator shall verify that the value of any (unwrapped) associated data, A, that was passed with the wrapped keying material is correct The evaluator shall feed into the TOE's KTS-KEM-KWS decryption operation examples of incorrect cipher text and verify that a decryption error is detected. If the key-wrapping algorithm is AES-CCM, the evaluator shall attempt at least one decryption where the wrong value of A is given to the KTS-KEM-KWS decryption operation and verify that a decryption error is detected. Similarly, if the key-wrapping algorithm is AES-CCM, the evaluator shall attempt at least one decryption where the wrong nonce is given to the KTS-KEM-KWS decryption operation and verify that a decryption error is detected.*
- ***Test 4:*** *The evaluator shall trigger all detectable decryption errors and validate that the resulting error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations (in particular, no Z values) are revealed to the sender.*

### FCS_COP.1(7) Cryptographic operation (Key Encryption)

FCS_COP.1.1(7)

The TSF shall **[selection:**

- ***not perform key encryption,***
- ***use platform-provided functionality to perform Key Wrapping,***
- ***perform [key encryption and decryption] in accordance with a specified cryptographic algorithm [AES used in CBC mode] and cryptographic key sizes [selection:***
  - ***128,***
  - ***256***

  ***] bits that meet the following: [AES as specified in SP 800-38A].***

**]**.

**Application Note:** This applies to any key encryption occurring on the enterprise server. This requirement is used in the body of the ST if the ST author chooses to use AES encryption/decryption for protecting the keys as part of the key chaining approach that is specified in FCS_KYC_EXT.1.

**Evaluation Activity**

***TSS***
*Conditional: If 'not perform key encryption' was selected, then the evaluator shall only examine the TSS to verify no key encryption is performed.*
*Requirement met by the platform If the platform provides the FEK encryption/decryption, then the evaluator shall examine the TSS to verify that it describes how the FEK encryption/decryption is invoked.*
*Requirement met by the TOE The evaluator shall verify the TSS includes a description of the key size used for encryption and the mode used for the key encryption*

***Guidance***
*None.*

***Tests***
*Conditional: If 'not perform key encryption' was selected, no testing is performed.*
*The assurance activity tests specified for AES in CBC mode in the underlying [AppPP] shall be performed in the case that "CBC" is selected in the requirement.*

### FCS_IV_EXT.1 Initialization Vector Generation

FCS_IV_EXT.1.1

The TSF shall [**selection**:

- *invoke platform-provided functionality to generate IVs,*
- *generate IVs with the following properties [**selection**:*
  - *CBC: IVs shall be non-repeating and unpredictable,*
  - *CCM: Nonce shall be non-repeating and unpredictable,*
  - *XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,*
  - *GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key*

  *]*

].

**Application Note:** This applies to any IV generation occurring on the enterprise server.

**Evaluation Activity**

***TSS***
*The evaluator shall ensure the TSS describes how nonces are created uniquely and how IVs and tweaks are handled (based on the AES mode). The evaluator shall confirm that the nonces are unique and the IVs and tweaks meet the stated requirements.*

***Guidance***
*None.*

***Tests***
*None.*

### FCS_KDF_EXT.1 Cryptographic Key Derivation Function

FCS_KDF_EXT.1.1

The TSF shall [**selection**:

- *not derive keys,*
- *accept [**selection**: a submask generated by an RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**, a conditioned password, an imported submask] to derive an intermediate key, as defined in [**selection**:*
  - *NIST SP 800-108 [**selection**: KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iteration Mode] ,*
  - *NIST SP 800-132*

  *] using the keyed-hash functions specified in FCS_COP.1**(4) (from [AppPP])**, such that the output is at least of equivalent security strength (in number of bits) to the [FEK(s)]*

].

**Application Note:** This applies to any key derivation occurring on the enterprise server. This requirement establishes acceptable methods for generating a new random key or an existing submask to create a new key along the key chain.

**Evaluation Activity**

***TSS***
*Conditional: If 'not derive keys' was selected, then the evaluator shall only examine the TSS to verify no key derivation is performed.*

*The evaluator shall verify the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 and SP 800-132.*

***KMD***

*The evaluator shall examine the vendor's KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived, including the input values. The evaluator shall confirm the input values are from the sources listed in the requirement. The evaluator will confirm the output is of equivalent strength to the FEK(s) it is protecting.*

***Guidance***
*None.*

***Tests***
*None.*

### FCS_KYC_EXT.1 Key Chaining and Key Storage

FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of [*[intermediate keys] originating from **one or more initial [selection: submask(s), recovery value(s)]** to [the final value returned to the endpoint] using the following method(s): [**selection**:*

- *utilization of the platform key storage,*
- *utilization of platform key storage that performs key wrap with a TSF provided key,*
- *implementation of key derivation as specified in FCS_KDF_EXT.1,*
- *implementation of key wrapping as specified in FCS_COP.1**(5)**,*
- *implementation of key combining as specified in FCS_SMC_EXT.1,*
- *implementation of key encryption as specified in FCS_COP.1**(7)**,*
- *implementation of key transport as specified in FCS_COP.1**(6)***

*] while maintaining an effective strength of [**selection**:*

- *[**selection**: 128 bits, 256 bits] for symmetric keys ,*
- *[**selection**: 128 bits, 192 bits, 256 bits] for asymmetric keys*

*]* commensurate with the strength of the FEK and [**selection**:

- *no supplemental key chains,*

- *other supplemental key chains that protect a key or keys in the primary key chain using the following method(s): [**selection**:*
  - *utilization of the platform key storage,*
  - *utilization of platform key storage that performs key wrap with a TSF provided key,*
  - *implementation of key wrapping as specified in FCS_COP.1**(5)**,*
  - *implementation of key combining as specified in FCS_SMC_EXT.1,*
  - *implementation of key encryption as specified in FCS_COP.1**(7)**,*
  - *implementation of key transport as specified in FCS_COP.1**(6)**,*
  - *implementation of key derivation as specified in FCS_KDF_EXT.1*

  *]*

].

**Application Note:** Key Chaining is the method of using multiple layers of encryption keys to ultimately secure the a final key. The number of intermediate keys will vary. The ST Author should clearly indicate which portions of the key chain are created and maintained by the enterprise server and which are created and maintained by the endpoint. This requirement is in addition to the same requirement in the File Encryption Module, it covers a different section of the keychain, if both modules are included both requirements must be included.

**Evaluation Activity**

***TSS***
*The evaluator shall verify the TSS contains a high-level description of the key sizes that it supports key outputs of no fewer 128 bits for products that support only AES128, and no fewer than 256 bits for products that support AES-256. The evaluator shall verify the TSS contains a description of the controls preventing a key from being provided to the endpoint before validation has occurred.*

*The evaluator shall verify the TSS includes a description of the key chain used to protect encryption keys associated with endpoints. The description of the key chains shall be reviewed to ensure it maintains a chain of keys using the methods listed in the SFR.*

*The evaluator shall ensure the chain of keys is maintained from the authorization factor or recovery value to the value returned to the endpoint. The evaluator shall examine the TSS to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. This description must include a diagram illustrating the key chain implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key chain to ensure that at no point the chain could be broken without a cryptographic exhaust, the initial authorization value, recovery value or a compromise of the TOE server and the effective strength of the keys are maintained throughout the key chain.*

***Guidance***
*If there are configurations to enable or disable use of enterprise server, which modify the key chain, they shall be described. If there are configurations on to enable recovery mechanisms, they shall be described.*

***Tests***
*None.*

### FCS_SMC_EXT.1 Submask Combining

FCS_SMC_EXT.1.1

The TSF shall [**selection**:

- *not perform submask combining,*
- *combine submasks using the following method [**selection**: exclusive OR (XOR), SHA-256, SHA-384, SHA-512, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] to generate an intermediate key*

].

**Application Note:** This applies to any submask combining occurring on the enterprise server. This requirement specifies the way that a product may combine the various submasks by using either an XOR or an approved SHA-hash.

**Evaluation Activity**

***TSS***
*Conditional: If 'not perform submask combining' was selected, then the evaluator shall only examine the TSS to verify no submask combining is performed.*

*If keys are XORed together to form an intermediate key, the TSS section shall identify how this is performed (e.g., if there are ordering requirements, checks performed, etc.). The evaluator shall also confirm that the TSS describes how the length of the output produced is at least the same as that of the FEK.*

***Guidance***
*None.*

***Tests***
*None.*

**FCS_VAL_EXT.1(1) Validation (Server Administrator)**

FCS_VAL_EXT.1.1(1)

The TSF shall perform validation of the [*admin*] by [**selection**:

- *receiving assertion of the subject's validity from [**assignment**: Operational Environment component responsible for authentication],*
- *validating the [**selection**: submask, intermediate key] using the following methods: [**selection**:*
  - *key wrap as specified in FCS_COP.1**(5)**,*
  - *hash the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(2) (from [AppPP])** and compare it to a stored hash,*
  - *decrypt a known value using the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(1) (from [AppPP])** and compare it against a stored known value*

  *]*

].

FCS_VAL_EXT.1.2(1)

The TSF shall require validation of the [*admin*] prior to [*permitting the actions described in FMT_MTD.1.1 and FMT_SMF.1.1(2)*].

**Application Note:** This PP-Module performs validation of any administrator credential on the management server, as described in FIA_AUT_EXT.1.1, used to log in to the EM in accordance with this SFR.

**Evaluation Activity**

***TSS***
*Conditional:*
*If 'validating' is selected, the evaluator shall examine the TSS to determine that it states which authorization factors support validation.*

*The evaluator shall also examine the TSS to ensure that it includes a high-level description of how how the submasks are validated. If multiple submasks are used within the TOE, the evaluator shall confirm that the TSS describes how each is validated (e.g., each submask validated before combining, once combined validation takes place).*

*Conditional:*
*If 'receiving assertion of the subject's validaity' is selected, the evaluator shall examine the TSS to verify that it describes the enviroments that can be leveraged with the TOE and how each claims to perform validation. The evaluator shall also ensure that none of the stated platform validation mechanisms weaken the key chain of the product.*

***Guidance***
*If the validation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established.*

***Tests***
*There are no test activities for this requirement.*

**FCS_VAL_EXT.1(2) Validation (User)**

FCS_VAL_EXT.1.1(2)

The TSF shall perform validation of the [*user*] by [**selection**:

- *receiving assertion of the subject's validity from [**assignment**: Operational Environment component responsible for authentication],*
- *validating the [**selection**: submask, intermediate key] using the following methods: [**selection**:*
  - *key wrap as specified in FCS_COP.1**(5)**,*
  - *hash the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(2) (from [AppPP])** and compare it to a stored hash,*
  - *decrypt a known value using the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(1) (from [AppPP])** and compare it against a stored known value*

  *]*

].

FCS_VAL_EXT.1.2(2)

The TSF shall require validation of the [*user*] prior to [*transmitting submasks, FEKs, or keys to decrypt FEKs to the endpoint*].

**Application Note:** This references the validation of an endpoint user to the server. These activities are performed by the server.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine which component of the*

*Operational Environment is used to assert the User's identity.*

*The evaluator shall examine the TSS to determine how the TOE responds to an assertion by the Operational Environment. The evaluator shall examine the TSS to verify that it describes how validation is performed. The evaluator shall verify the TSS ensures that the validation process does not expose any material that might compromise key material or expose protected data.*

***Guidance***
*The evaluator shall examine the operational guidance to ensure it describes how to configure the TOE and Operational Environment to enable the OE to provide User identity assertions to the TOE.*

*(conditional) If the number of User authentication attempts is configurable in the TOE, the examiner shall examine the operational guidance to ensure it describes how to configure the TOE.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any "lockout" period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.*
- ***Test 2:*** *For each validated authorization factor, ensure that when the user provides an incorrect authorization factor, the TOE prevents FEKs or keys that decrypt FEKs from being forwarded to the endpoint.*

#### FCS_VAL_EXT.2(2) Validation Remediation (User)

FCS_VAL_EXT.2.1(2)

The TSF shall [**selection**:

- *[[**selection**: direct the endpoint to perform key sanitization, perform key sanitization] of FEK(s) or an intermediate key] upon **[assignment: ST specified number or configurable range of]** consecutive failed validation attempts,*
- *institute a delay such that only [**assignment**: ST author specified number **or configurable range** of attempts] can be made within a 24 hour period,*
- *block validation after [**assignment**: ST author specified number **or configurable range** of attempts] of consecutive failed validation attempts,*
- ***terminate the session** after [**assignment**: ST author specified number **or configurable range** of attempts] of consecutive failed validation attempts*

].

**Evaluation Activity**

***TSS***
*This SFR is evaluated through the activities defined for FCS_VAL_EXT.1(2).*

***Guidance***
*This SFR is evaluated through the activities defined for FCS_VAL_EXT.1(2).*

***Tests***
*This SFR is evaluated through the activities defined for FCS_VAL_EXT.1(2).*

### 5.2.2 Identification and Authentication (FIA)

#### FIA_AUT_EXT.1 Subject Authorization

FIA_AUT_EXT.1.1

The TSF shall [**selection**: *receive assertion of the user's validity from: [**assignment**: Operational Environment component responsible for user authentication], provide authorization*] based on [**selection**:

- *a password authorization factor conditioned as defined in FCS_CKM_EXT.6,*
- *an external smart card factor that is at least the same bit-length as the FEK(s), and is protecting a submask that is [**selection**: generated by the TOE (using the RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**), generated by the platform] protected using RSA with key size [**selection**: 3072 bits, 4096 bits] with user presence proved by presentation of the smart card and [**selection**: no PIN, an OE defined PIN, a configurable PIN]*
  ,
- *an external USB token factor that is at least the same security strength as the FEK(s), and is providing a submask generated by the [**selection**: TOE, using the RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**, platform]*

].

**Application Note:** This applies to the authorization of administrators on the enterprise server.

FCS_RBG_EXT.1 is in the Application Software Protection Profile.

This requirement specifies what authorization factors the TOE accepts from the user. A password entered by the user is one authorization factor that the TOE must be able to condition, as specified in FCS_CKM_EXT.6. Another option is a smart card authorization factor, with the differentiating feature being how the value is generated – either by the TOE's RBG or by the platform. An external USB token may also be used, with the submask value generated either by the TOE's RBG or by the platform.

The TOE may accept any number of authorization factors, and these are categorized as "submasks". The ST author selects the authorization factors they support, and there may be multiple methods for a selection.

Use of multiple authorization factors is preferable; if more than one authorization factor is used, the submasks produced must be combined using FCS_SMC_EXT.1.

**Evaluation Activity**

*The evaluation activities for this component will be driven by the selections made by the ST author. This section describes evaluation activities for all possible selections in an ST; it should be understood that if a capability is not selected in the ST, the noted assurance activity does not need to be performed.*

***TSS***
*The evaluator shall examine the TSS to ensure that it describes how user authentication is performed. The evaluator shall verify that the authorization methods listed in the TSS are specified and included in the requirements in the ST.*

*Requirement met by the TOE:*
*The evaluator shall first examine the TSS to ensure that the authorization factors specified in the ST are described. For password-based factors the examination of the TSS section is performed as part of FCS_CKM_EXT.6 Evaluation Activities. Additionally in this case, the evaluator shall verify that the operational guidance discusses the characteristics of external authorization factors (e.g., how the authorization factor must be generated; format(s) or standards that the authorization factor must meet) that are able to be used by the TOE.*

*If other authorization factors are specified, then for each factor, the TSS specifies how the factors are input into the TOE.*

*Requirement met by the OE:*
*The evaluator shall examine the TSS to ensure a description is included for how the TOE is invoking the OE functionality and how it is getting an authorization value that has appropriate entropy.*

***Guidance***
*The evaluator shall verify that the AGD guidance includes instructions for all of the authorization factors. The AGD will discuss the characteristics of external authorization factors (e.g., how the authorization factor is generated; format(s) or standards that the authorization factor must meet, configuration of the TPM device used) that are able to be used by the TOE.*

***Tests***
*The evaluator shall ensure that authorization using each selected method is tested during the course of the evaluation, setting up the method as described in the operational guidance and ensuring that authorization is successful and that failure to provide an authorization factor results in denial to access to plaintext data.*
*[conditional]: If there is more than one authorization factor, ensure that failure to supply a required authorization factor does not result in access to the decrypted plaintext data.*

### FIA_REC_EXT.1 Recovery Support

FIA_REC_EXT.1.1

The TSF shall [**selection**: *provide the ability to enable and disable the use of recovery credentials*, *not support recovery*].

FIA_REC_EXT.1.2

The TSF shall support the following recovery mechanisms [**selection**: *Challenge Response Recovery as defined in FIA_CHR_EXT.1*, *None*].

**Application Note:** This requirement defines the recovery options supported between the endpoint(s) and the enterprise server. This does not prevent the OE from providing recovery if the OE is managing the authentication of the users.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that types of supported recovery credential are specified.*

***Guidance***
*The evaluator shall confirm that the guidance documentation contains instructions for turning off the ability of the server to return a recovery credential.*

***Tests***
*The evaluator shall disable the ability of a server to return a recovery credential. The evaluator should then attempt to obtain the recovery credential and this should fail.*

**FIA_UAU.1 Timing of Authentication**

FIA_UAU.1.1

The TSF shall allow [**assignment**: *list of TSF-mediated actions*] on behalf of the **administrator** to be performed before the administrator is authenticated.

FIA_UAU.1.2

The TSF shall require each **administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that **administrator**.

**Application Note:** This requirement defines the timing of administrator capabilities on the enterprise server.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that it describes the list of actions that are performed on behalf of the administrator prior to login of the administrator. The evaluator shall examine the TSS to determine that it describes the list of actions that require administrator authentication.*

***Guidance***
*The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre- shared keys, tunnels, certificates, etc.) to logging in are described. For each supported login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall verify that the list of actions allowed without administrator login completes successfully without requiring administrator login and make sure this list is consistent with the TSS.*
- ***Test 2:*** *The evaluator shall verify that attempting any other action requires successful entry of an administrator credential.*
- ***Test 3:*** *The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.*
- ***Test 4:*** *The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.*

**FIA_UID.1 Timing of Identification**

FIA_UID.1.1

The TSF shall allow [**assignment**: *list of TSF-mediated actions*] on behalf of the administrator to be performed before the **administrator** is identified.

FIA_UID.1.2

The TSF shall require each **administrator** to be successfully identified before allowing any other TSF-mediated actions on behalf of that **administrator**.

**Application Note:** This requirement defines the timing of administrator capabilities on the enterprise server.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that it describes the list of actions that are performed on behalf of the administrator prior to identification of the administrator.*

***Guidance***
*The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps for creating and configuring administrator accounts are described.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall verify that the list of actions allowed without administrator identification completes successfully without requiring the administrator to be identified and make sure this list is consistent with the TSS.*
- ***Test 2:*** *The evaluator shall verify that attempting any other action requires successful entry of an administrator account name and successful entry of the administrator account credential.*

### 5.2.3 Security Management (FMT)

#### FMT_MOF.1 Server Management of Security Functions Behavior

FMT_MOF.1.1

The TSF shall restrict the ability to [**selection**: *determine the behavior of, disable, enable, modify the behavior of*] the functions **[selection: *encryption algorithms used, key sizes used*]** to [*administrators*].

**Application Note:** The intent of this SFR is to define a mechanism to distinguish administrators (who have the ability to configure the TSF and its data) from users (individuals in the enterprise who have FEs on their systems).

The TSF does not need to provide roles that are explicitly called ‘administrator’ or ‘user’; the ST must logically define the administrator as a combination of one or more roles that are provided by the TOE. A user as defined by this PP-Module may be either a user that is specifically assigned an unprivileged role by the TSF or it may be characterized by an individual that lacks an administrator account on the TOE

The TSF may optionally provide the ability to rely on an external authentication mechanism to identify users in the case of a user requesting distribution of a recovery credential. In this situation, the TOE’s reliance on the Operational Environment is functionally equivalent to the TSF maintaining the user role as defined by FMT_SMR.2.1.

**Evaluation Activity**

***TSS***
*The evaluator shall examine that the TSS details how Administrators are authenticated and identified by all TOE components. The evaluator shall examine that authentication and identification of Administrators cannot be compromised for any TOE component in this case.*

***Guidance***
*The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.*

***Tests***
*The evaluator shall perform the following tests:*
*In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP-Module be tested; for instance, if the TOE can be administered through a local hardware interface, SSH, and TLS/HTTPS, then all three methods of administration must be exercised during the evaluation team’s test activities.*

#### FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1

The TSF shall restrict the ability to [**selection**: *change default, query, modify, delete, clear, [**assignment**: other operations]*] the [*encryption keys and intermediate values*] to [*administrators*] **at the following times: [selection: *never, during initial provisioning, during recovery*]**.

**Application Note:** These restrictions apply to modifications on the enterprise server.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are available to the administrator are identified. For each of these functions, the evaluator shall also confirm that the TSS details when changes may be made to the encryption keys and/or intermediate values.*

***Guidance***
*The evaluator shall verify that the guidance document describes what operations on the encryption keys and intermediate values are allowed to the administrator at what times.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall try to perform at least one of the related actions without prior authentication as administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). This test should fail.*
- ***Test 2:*** *The evaluator shall try to perform at least one of the related actions with prior authentication as administrator. This test should pass.*
- ***Test 3:*** *The evaluator shall try to perform at least one of the actions at the times that are not permitted. This test should fail.*
- ***Test 4:*** *The evaluator shall try to perform at least one of the actions at the times are permitted. This test should pass.*

**FMT_SMF.1(2) Specification of Management Functions (Management Server)**

FMT_SMF.1.1(2)

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

- ***register new user,***
- ***revoke registration of user,***
- ***initiate key generation,***
- ***initiate key escrow,***
- ***initiate key zeroization,***
- ***initiate key recovery,***
- ***set encryption policy (supported algorithms and key sizes),***
- ***change administrator passwords,***
- ***change user passwords,***
- ***change recovery credentials,***
- ***define administrators of the TOE,***
- ***enable/disable use of recovery credential,***
- ***configure number of failed authentication attempts before issuing a key destruction of the FEK(s),***
- ***configure the number of authentication attempts that can be made within a 24 hour period,***
- ***configure the number of failed authentication attempts required to begin blocking subsequent attempts,***
- ***ability to enable or disable one or more of the following functions: [selection: configure cryptographic functionality, change authentication factors, perform a cryptographic erase of the data by the destruction of FEKs or KEKs protecting the FEKs, configure the number of failed validation attempts required to trigger corrective behavior, configure the corrective behavior to issue in the event of an excessive number of failed validation attempts, [assignment: other management functions provided by the TSF]]***

**]**.

**Application Note:** This SFR refers specifically to the management functions that can be performed by the Management Server. Functions that are performed by the rest of the TOE are addressed by the FMT_SMF.1(2) SFR in the File Encryption PP-Module. The final two assignments provide the ST author the ability to indicate when File Encryption module functionality (such as configuration of power saving states) can be configured by the Management Server.

The TSF's ability to initiate key generation, escrow, zeroization, and/or recovery may be accomplished either by the TOE performing those functions or by the TOE issuing a request to a remote client to perform the functions. The ST author must indicate which case is provided by the TSF. If the TOE performs any of the cryptographic functions that are selected as being initiated in this SFR, the ST author must include the equivalent FCS SFRs from the File Encryption PP-Module as part of the TOE, specifically indicating that these functions are provided by the Management Server component of the TOE.

If the TSF supports the use of a recovery credential, the ST author must include the 'enable/disable use of recovery credential' selection.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to ensure that it describes which of the selections are provided by the TOE. Additionally, the TSS shall describe which of the configurable selections can be disabled on the Enterprise Management Server. The evaluator shall examine the TSS to ensure that it describes whether the TOE provides the ability to initiate key generation, escrow, zeroization and/or recovery or whether it requests the client to perform those functions.*

***Guidance***
*The evaluator shall examine the Guidance Documents to ensure that, if supported, configuration of the following options is described, including any reliance on the Operational Environment if applicable:*

- *Register new user*

- *Revoke registration of an user*
- *Initiate key generation*
- *Initiate key escrow*
- *Initiate key recovery*
- *Initiate key zeroization*
- *Set encryption policy (supported algorithms and key sizes)*
- *Change Administrator passwords*
- *Change user passwords*
- *Change Recovery Credentials*
- *Define Administrators of the TOE*
- *Enable/Disable the use of recovery credentials (end users)*
- *Configure the number of failed authentication attempts before issuing a key destruction of the FEK(s)*
- *Configure the number of authentication attempts that can be made in a 24 hour period*
- *Configure the number of failed authentication attempts required to begin blocking subsequent attempts*
- *The ability to enable/disable one or more functions defined in the File Encryption module*
- *The ability to authorize whether or not users can perform one or more of the functions in the File Encryption PP-Module.*
- *ability to enable or disable one or more of the following functions (configure cryptographic functionality, change authentication factors, perform a cryptograph erase of the data by the destruction of FEKs or KEKs protecting the FEKs, configure the number of failed validation attempts required to trigger corrective behavior, configure the corrective behavior to issue in the event of an excessive number of failed validation attempts, [other management functions provided by the TSF]*
- *ability to perform one or more of the following functions (configure cryptographic functionality, change authentication factors, perform a cryptograph erase of the data by the destruction of FEKs or KEKs protecting the FEKs, configure the number of failed validation attempts required to trigger corrective behavior, configure the corrective behavior to issue in the event of an excessive number of failed validation attempts, [other management functions provided by the TSF]*
- *ability to authorize whether or not users can perform one or more of the following functions (configure cryptographic functionality, change authentication factors, perform a cryptograph erase of the data by the destruction of FEKs or KEKs protecting the FEKs, configure the number of failed validation attempts required to trigger corrective behavior, configure the corrective behavior to issue in the event of an excessive number of failed validation attempts, [other management functions provided by the TSF]*

***Tests***

*The evaluator shall perform the following tests for each claimed management function:*

- ***Test 1:*** *The evaluator shall configure the management server and two users according to the guidance documents. The evaluator shall register the users with the management server. The evaluator shall verify that the users are identified by the management server as defined in the guidance documents. This test shall pass.*
- ***Test 2:*** *The evaluator shall disconnect the second user from the network. The evaluator shall revoke the registration of the second user in the management server. The evaluator shall attempt to connect the second user to the network and verify the endpoint fails to connect or is displayed as revoked in the console.*
- ***Test 3:*** *The evaluator shall verify that the TOE performs the actions (e.g. generate key) and sends the result to the user's client. The user's client shall perform the actions necessary to accept the updated configuration (e.g. encrypt the data with the new key, update the encryption algorithm key size or mode and re-encrypt).*
- ***Test 4:*** *For each item that is initiated by the TOE but performed on the endpoint, the evaluator shall verify that the TOE requests the user's client to perform the action (generate a key and encrypt the data, zeroize a key).*
- ***Test 5:*** *For each method of changing a credential, the evaluator shall first provision the initial authorization factor(s) in the Enterprise Server, and then verify all authorization values supported allow the user access to the encrypted data on the user's client. Then the evaluator shall exercise the management functions to change the authorization factor values to a new one on the Enterprise Server. Then he or she will verify that the user's client denies access to the user’s encrypted data when he or she uses the old or original authorization factor values to gain access.*
- ***Test 6:*** *The evaluator shall add two administrators to the administrator group in the Enterprise Server and provision authorization factor(s) for each administrator. The evaluator shall verify that both administrators can log into the Enterprise Server using the provided authorization factors. The evaluator shall then exercise the management functions to change the authorization factor values for the first administrator to a new one on the Enterprise Server. Then he or she will verify that the Enterprise Server denies the first administrator access to the Management Console when the first administrator logs in with the old or original authorization factor to gain access. The evaluator shall also verify that the second administrator is still able to log in to the Enterprise Server with their original authorization factor.*
- ***Test 7:*** *The evaluator shall verify that the an administrator can configure each of the supported authorization factors attempts limits*

*and shall verify that the user is denied access after surpassing that limit.*

- ***Test 8:*** *If the TOE provides the capability to disable management of any capability allowed in the EM PP-Module, the evaluator shall devise a test that ensures that each capability which can be disabled has been or can be disabled following guidance provided by the vendor.*
- ***Test 9:*** *If the TOE provides the capability to manage capabilities in place of the File Encryption Clients, where those administrative capabilities are then disabled in the File Encryption Clients, the evaluator shall devise a test that ensures that each capability which can be disabled in the File Encryption Clients and can be subsequently managed by the EM is tested as follows: Disable the administrative capability in a File Encryption Client and enable it in the EM. Verify that the administration of the capability in the EM is successful.*
- ***Test 10:*** *The evaluator shall verify the encryption policy enforcement by changing the permitted algorithms and verifying the changes take place.*

#### FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1

The TSF shall maintain the roles [*administrator, user*].

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions **[selection: *the administrator role shall be able to administer the Management Server locally , the administrator role shall be able to administer the Management Server remotely as specified in FTP_TRP.1, the administrator role shall be able to administer the endpoint(s) locally, the administrator role shall be able to administer the endpoint(s) remotely*]** are satisfied.

**Application Note:** The intent of this SFR is to define a mechanism to distinguish administrators (who have the ability to configure the TSF and its data) from users (individuals in the enterprise who have FEs on their systems).

The TSF does not need to provide roles that are explicitly called ‘administrator’ or ‘user’; the ST must logically define the administrator as a combination of one or more roles that are provided by the TOE. A user as defined by this PP-Module may be either a user that is specifically assigned an unprivileged role by the TSF or it may be characterized by an individual that lacks an administrator account on the TOE.

The TSF may optionally provide the ability to rely on an external authentication mechanism to identify users in the case of a user requesting distribution of a recovery credential. In this situation, the TOE’s reliance on the Operational Environment is functionally equivalent to the TSF maintaining the user role as defined by FMT_SMR.2.1.

**Evaluation Activity**

***TSS***
*Refer to the evaluation activities for FMT_MOF.1.*

***Guidance***
*Refer to the evaluation activities for FMT_MOF.1.*

***Tests***
*Refer to the evaluation activities for FMT_MOF.1.*

### 5.2.4 Protection of the TSF (FPT)

#### FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1

The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE through the use of **[selection: *IPsec as defined in the PP-Module for VPN Client, HTTPS in accordance with FCS_HTTPS_EXT.1 (from [AppPP]), TLS as defined in the Package for Transport Layer Security, SSH as defined in the Extended Package for Secure Shell*]**.

**Application Note:** This SFR is intended to define protected communications between the Management Server and the endpoints.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols and intra-TOE configurations for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.*

***Guidance***
*The evaluator shall confirm that the guidance documentation contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains instructions to reestablish a connection should a connection be unintentionally broken.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall ensure that communications using each supported protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.*
- ***Test 2:*** *The evaluator shall ensure, for each communication channel with an endpoint or server, the channel data is not sent in plaintext.*
- ***Test 3:*** *The evaluator shall, for each protocol associated with each authorized IT entity tested during Test 1, physically interrupt the connection. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.*

*Further evaluation activities are associated with the specific protocols.*

### FPT_KYP_EXT.1 Protection of Keys and Key Material

FPT_KYP_EXT.1.1

The TSF shall store keys in non-volatile memory only when [**selection**:

- *wrapped, as specified in FCS_COP.1(5),*
- *encrypted, as specified in FCS_COP.1(1)* ***(from [AppPP])****,*
- *the plaintext key is stored in the underlying platform's keystore as specified by FCS_STO_EXT.1.1* ***(from [AppPP])****,*
- *the plaintext key is stored in a SQL database in the Operational Environment,*
- *the plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1.,*
- *the plaintext key will no longer provide access to the encrypted data after initial provisioning,*
- *the plaintext key is a key split that is combined as specified in FCS_SMC_EXT.1 and another contribution to the split is [**selection**: wrapped as specified in FCS_COP.1(5) or encrypted as specified in FCS_COP.1(7), derived and not stored in non-volatile memory] ,*
- *the plaintext key is stored on an external storage device for use as an authorization factor.,*
- *the plaintext key is used to encrypt a key as specified in FCS_COP.1(7) or wrap a key as specified in FCS_COP.1(5) that is already encrypted as specified in FCS_COP.1(7) or wrapped as specified in FCS_COP.1(5)*

].

**Application Note:** This details the key storage requirements for the enterprise server. The plaintext key storage in non-volatile memory is allowed for several reasons. If the keys exist within protected memory that is not user accessible on the TOE or OE, the only methods that allow it to play a security relevant role for protecting the FEK is if it is a key split or providing additional layers of wrapping or encryption on keys that have already been protected.

**Evaluation Activity**

***TSS***
*The evaluator shall verify the TSS for a high level description of the method(s) used to protect keys stored in non-volatile memory.*

***KMD***

*The evaluator shall verify the KMD to ensure it describes the storage location of all keys and the protection of all keys stored in non-volatile memory. The description of the key chain shall be reviewed to ensure FCS_COP.1(5) is followed for the storage of wrapped or encrypted keys in non-volatile memory and plaintext keys in non-volatile memory meet one of the criteria for storage.*

***Guidance***
*None.*

***Tests***
*None.*

### FPT_KYP_EXT.2 Attribution of Key and Key Material

FPT_KYP_EXT.2.1

The TSF shall maintain an association between stored endpoint keys and user identity, [**selection**: *remote endpoints*, *recovery credential*, *system identity*, *no other subjects*].

**Application Note:** The intent of this SFR is that at minimum, keys are associated with the users for which it was explicitly created by the TSF. If the

TOE has the ability to maintain an association to keys for a user, this SFR is intended to require an association between the key chain and a user through the user account name(s) that are authorized to use it.

Likewise, if the TOE supports the use of a recovery credential, this SFR is intended to require an association between user and the recovery credential used to recover that data.

FPT_KYP_EXT.2.2

The TSF shall provide the ability to register users by exchange of [**assignment**: *mutually identifying information that allows for an association to be made*].

**Application Note:** The ST author will complete the assignment with information on the method used by the Management Server portion of the TOE to establish the association with the endpoint portion of the TOE described in FPT_KYP_EXT.2.1.

FPT_KYP_EXT.2.3

The TSF shall provide the ability to revoke the registration of users by [**assignment**: *method of removing and/or exchanging information that prevents further communications between the TOE and the endpoint*].

FPT_KYP_EXT.2.4

The TSF shall transmit any secure or private cryptographic information that is transferred between the TOE and a user's endpoint in order to establish or disestablish an association using a communications channel with a security strength at least as great as the strength of the information being transmitted.

**Application Note:** The channel used to transmit this data is defined in FPT_ITT.1.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to verify that it describes the method by which an association is maintained and verify it matches the selections.*

***Guidance***
*The evaluator shall verify the guidance documentation provides instructions on how to configure the association, if any configuration is necessary.*

***Tests***
*For each method of association, the evaluator shall change the configuration so that the associate is broken and verify that enterprise functions do not work.*

# 6 Consistency Rationale

## 6.1 Application Software Protection Profile

### 6.1.1 Consistency of TOE Type

When 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 simply extended to include the enterprise management functionality for software file encryption that the application performs.

### 6.1.2 Consistency of Security Problem Definition

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

| PP-Module Threat | Consistency Rationale |
|---|---|
| T.KEYING_MATERIAL_COMPROMISE_SERVER | This threat is a specific example of T.PHYSICAL_ACCESS defined in the Base-PP. Specifically, this PP-Module defines a method of maliciously gaining access to sensitive data at rest that is particular to the technology type of this PP-Module. |
| T.MAN_IN_THE_MIDDLE | This threat is a specific example of T.NETWORK_EAVESDROP defined in the Base-PP. Specifically, the attacker performs network eavesdropping to gain access to key data in transit between TOE components. |
| T.UNAUTHORIZED_ADMINISTRATOR_ACCESS | This threat is a variation on T.LOCAL_ATTACK defined in the Base-PP. The Base-PP does not define access-controlled management functions so this PP-Module goes beyond it by specifying misuse of the management interface as a threat to the TSF. |
| T.UNTRUSTED_COMMUNICATION_CHANNELS | This threat is a variation on T.NETWORK_ATTACK and T.NETWORK_EAVESDROP defined in the Base-PP. The threat of untrusted communication channels allows for exploitation of the TSF in different ways, depending on how the lack of trust is manifested. |
| T.UNAUTHORIZED_DATA_ACCESS_ENDPOINT | This threat is a variation on T.PHYSICAL_ACCESS defined in the Base-PP. In this case, the "sensitive data at rest" is the data that the TOE is intended to protect. |
| T.UNAUTHORIZED_DATA_ACCESS_SERVER | This threat is a variation on T.PHYSICAL_ACCESS defined in the Base-PP. In this case, the "sensitive data at rest" is the data that the TOE is intended to protect. |

### 6.1.3 Consistency of Objectives

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

| PP-Module TOE Objective | Consistency Rationale |
|---|---|
| O.ENTERPRISE_KEY_PROTECTION | This objective is consistent with the Base-PP because the Base-PP includes the O.PROTECTED_STORAGE objective. The protection and timely destruction of key materials is consistent with the intent of that objective. |
| O.KEY_MATERIAL_SERVER | This objective is consistent with the Base-PP because the Base-PP includes the O.PROTECTED_STORAGE and O.PROTECTED_COMMS objectives. This objective defines behavior for the secure storage and transmission of decryption and recovery key data, consistent with the relevant objectives in the Base-PP. |
| O.RECOVERY_PROTECTION | This objective defines usage restrictions and supported behavior for the TOE's management interface. This is consistent with the O.MANAGE objective in the Base-PP for functionality that is specific to this PP-Module. |
| O.SECURE_CHANNEL | This objective is consistent with the Base-PP because the Base-PP defines the O.PROTECTED_COMMS objective for security of data in transit. Specifically, this objective expects the communications between distributed TOE components to be protected in a similar manner to the corresponding Base-PP objective. |
| O.VERIFIED_ADMIN | This objective is consistent with the O.MANAGE objective in the Base-PP and further restricts it by limiting security-relevant management interfaces to authenticated administrators. It also supports the enforcement of the OE.PROPER_ADMIN environmental objective by reducing the likelihood that administrative actions are performed unintentionally. |

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

| PP-Module Operational Environment Objective | Consistency Rationale |
|---|---|

| | |
|---|---|
| OE.ENVIRONMENTAL_STORAGE | This objective is consistent with the Base-PP because the Base-PP allows for the TOE to use platform-provided key storage. |
| OE.PHYSICAL_SERVER | This objective is consistent with the Base-PP because it is an extension of the Base-PP's OE.PLATFORM objective that is specific to this technology type. It is also consistent because the Base-PP permits the TSF to use platform-provided cryptography. |
| OE.SECURED_CONFIGURATION | This objective is consistent with the Base-PP because it expects the TOE's operational guidance to be responsibly followed in the same manner as OE.PROPER_ADMIN in the Base-PP. |
| OE.SECURED_ENVIRONMENT | This objective is consistent with the Base-PP because it is an extension of the Base-PP's OE.PLATFORM objective that is specific to this technology type. |

### 6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the App PP that are needed to support File Encryption Enterprise Management functionality. This is considered to be consistent because the functionality provided by the App is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the App PP as well as new SFRs that are used entirely to provide functionality for File Encryption Enterprise Management. 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** |
| FTP_DIT_EXT.1 | This SFR is defined in the Base-PP. This PP-Module modifies it by removing the option not to transmit sensitive data because this particular TOE type will always have that capability. It is still consistent with the Base-PP because all selections that the ST author is permitted to make are available options in the Base-PP version of the SFR. |
| | **Mandatory SFRs** |
| FCS_CKM_EXT.4 | This SFR extends the cryptographic functionality defined in the Base-PP by specifying a method for key destruction. It is consistent with the Base-PP because keys generated by the Base-PP portion of the TOE may also be destroyed in the manner specified by this SFR. |
| FCS_COP.1(5) | This SFR defines usage of AES functionality not defined by the Base-PP. However, this functionality is only used in certain situations that are specific to this PP-Module and do not affect the ability of any Base-PP SFRs to be enforced. |
| FCS_COP.1(6) | This SFR defines key transport functionality that is outside the scope of the original cryptographic operations defined in the Base-PP. |
| FCS_COP.1(7) | This SFR defines key encryption functionality that is outside the scope of the original cryptographic operations defined in the Base-PP. |
| FCS_IV_EXT.1 | This SFR defines how IVs for AES keys must be generated. This is consistent with the Base-PP because it supplements the key generation methods specified by the Base-PP SFR FCS_CKM.1(2). |
| FCS_KDF_EXT.1 | This SFR defines key transport functionality. It uses random bit generation and keyed-hash message authentication functionality from the Base-PP as they are intended but is otherwise outside the scope of the original cryptographic operations defined in the Base-PP. |
| FCS_KYC_EXT.1 | The Base-PP defines how stored keys are protected. This SFR extends that functionality by defining the logical hierarchy of how keys are logically protected by other keys or other secret data. |
| FCS_SMC_EXT.1 | This SFR relates to submask combining as a method of generating intermediate keys. Key hierarchy functionality is outside the scope of the Base-PP. |
| FCS_VAL_EXT.1(1) | This SFR goes beyond the functionality defined by the Base-PP by defining a method by which the TSF can validate the correctness of data input to it. |
| FCS_VAL_EXT.1(2) | This SFR goes beyond the functionality defined by the Base-PP by defining a method by which the TSF can validate the correctness of data input to it. |
| FCS_VAL_EXT.2(2) | This SFR goes beyond the functionality defined by the Base-PP by defining a method by which the TSF can take security-relevant action if some data input to it is invalid. |
| FIA_AUT_EXT.1 | This SFR defines how administrator requests to access protected data are authorized. It uses FCS_RBG_EXT.1 from the Base-PP in a manner consistent with its definition, but otherwise does not relate to functionality defined by the Base-PP. |
| FIA_REC_EXT.1 | This SFR defines the TOE's potential support for recovery credentials. This functionality does not relate to any behavior defined in the Base-PP. |
| FIA_UAU.1 | This SFR requires administrators to be authenticated prior to accessing management functionality. The Base-PP does not mandate identification and authentication measures for a management interface but it also does not prohibit them. |
| FIA_UID.1 | This SFR requires administrators to be identified prior to accessing management functionality. The Base-PP does not mandate identification and authentication measures for a management interface but it also does not prohibit them. |
| FMT_MOF.1 | This SFR defines access restrictions for TOE management functions. This is not |

| | |
|---|---|
| | defined in the Base-PP but there is nothing in the Base-PP that prohibits it. |
| FMT_MTD.1 | This SFR defines access restrictions for management of TSF data. This is not defined in the Base-PP but there is nothing in the Base-PP that prohibits it. |
| FMT_SMF.1(2) | This SFR defines management functions for the TOE for functionality specific to this PP-Module. These functions are defined in addition to what the Base-PP defines for its own operation. |
| FMT_SMR.2 | This SFR defines administrative roles, which are used by other SFRs to derive privileges to interact with the TOE's management functionality. This is not defined in the Base-PP but there is nothing in the Base-PP that prohibits it. |
| FPT_ITT.1 | This SFR uses a subset of the protocols defined in the Base-PP for secure communications. This PP-Module extends the functionality by explicitly defining a communications channel where both endpoints are TOE components. |
| FPT_KYP_EXT.1 | The Base-PP defines an SFR for secure storage of sensitive data. This SFR expands on that definition by describing the supported logical methods for storage of key data. |
| FPT_KYP_EXT.2 | This SFR relates to key attribution such that stored keys can be associated with the users that 'own' them. This does not relate to functionality that is defined in the Base-PP so it does not interfere with the implementation of any Base-PP SFRs. |
| **Optional SFRs** | |
| This PP-Module does not define any optional requirements. | |
| **Selection-based SFRs** | |
| FCS_CKM_EXT.6 | This SFR defines a key derivation method based on passphrase conditioning. It uses the FCS_RBG_EXT.1 SFR from the Base-PP in its intended manner but otherwise does not relate to the Base-PP's functionality. |
| FCS_VAL_EXT.2(1) | This SFR goes beyond the functionality defined by the Base-PP by defining a method by which the TSF can take security-relevant action if some data input to it is invalid. |
| FIA_CHR_EXT.1 | This SFR defines the TOE's implementation of recovery credentials. This functionality does not relate to any behavior defined in the Base-PP. |
| FTP_TRP.1 | This SFR uses a subset of the protocols defined in the Base-PP for secure communications. This PP-Module extends the functionality by explicitly defining a communications path between a remote administrator and the TOE. |
| **Objective SFRs** | |
| This PP-Module does not define any objective requirements. | |

# Appendix A - Optional SFRs

This PP-Module does not define any optional SFRs.

# Appendix B - Selection-based SFRs

**FCS_CKM_EXT.6 Cryptographic Password/Passphrase Conditioning**

> ***This is a selection-based component. Its inclusion depends upon selection from FIA_AUT_EXT.1.1.***

FCS_CKM_EXT.6.1

The TSF shall support a password/passphrase of up to [**assignment**: *maximum password size, positive integer of 64 or more*] characters used to generate a password authorization factor.

FCS_CKM_EXT.6.2

The TSF shall allow passwords 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*].

FCS_CKM_EXT.6.3

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*] bits that meet the following: [*NIST SP 800-132*].

FCS_CKM_EXT.6.4

The TSF shall not accept passwords less than [**selection**: *a value settable by the administrator, [**assignment**: minimum password length accepted by the TOE, must be >= 1]*] and greater than the maximum password length defined in FCS_CKM_EXT.6.1.

FCS_CKM_EXT.6.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 **(from [AppPP])** and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM_EXT.6.3.

**Application Note:** This applies to passwords on the enterprise server. The password/passphrase is represented on the host machine as a sequence of characters whose encoding depends on the TOE and the underlying OS. This sequence must be conditioned into a string of bits that is to be used as a KEK that is the same size as the FEK.

For FCS_CKM_EXT.6.1, the ST author assigns the maximum size of the password/passphrase it supports; it must support at least 64 characters.

For FCS_CKM_EXT.6.2, the ST author assigns any other supported characters; if there are no other supported characters, they should select “no other characters”.

For FCS_CKM_EXT.6.3, the ST author selects the parameters based on the PBKDF used by the TSF. The key cryptographic key sizes in FCS_CKM_EXT.6.3 are made to correspond to the KEK key sizes selected in FCS_KYC_EXT.1.

The password/passphrase must be conditioned into a string of bits that forms the submask to be used as input into the KEK. Conditioning is performed using one of the identified hash functions in accordance with the process described in NIST SP 800-132. SP 800-132 requires the use of a pseudo-random function (PRF) consisting of HMAC with an approved hash function.

Appendix A of SP 800-132 recommends setting the iteration count in order to increase the computation needed to derive a key from a password and, therefore, increase the workload of performing a password recovery attack. However, for this PP-Module, a minimum iteration count of 4096 is required in order to ensure that twelve bits of security is added to the password/passphrase value. A significantly higher value is recommended to ensure optimal security.

For FCS_CKM_EXT.6.4 If the minimum password length is settable, then ST author chooses “a value settable by the administrator for this component for FMT_SMF.1.1(2). If the minimum length is not settable, the ST author fills in the assignment with the minimum length the password must be (zero-length passwords are not allowed for compliant TOEs).

This requirement is selection dependent on FIA_AUT_EXT.1.1.

**Evaluation Activity**

> ***TSS***
> *There are two aspects of this component that require evaluation: passwords/passphrases of the length specified in the requirement (at least 64 characters) are supported, and that the characters that are input are subject to the selected conditioning function. These activities are separately addressed in the text below.*
>
> *Support for minimum length: The evaluator shall check to ensure that the TSS describes the allowable ranges for password/passphrase lengths, and that at least 64 characters 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 formation of all KEKs or FEKs (as decided in the FCS_CKM_EXT.3 selection) 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 password/passphrase is first encoded and then fed to the SHA 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. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length of the KEK selected in FCS_KYC_EXT.1.*

*For the NIST SP 800-132-based conditioning of the password/passphrase, the required evaluation activities will be performed when doing the evaluation activities for the appropriate requirements (FCS_COP.1.1(4)). If any manipulation of the key is performed in forming the submask that will be used to form the FEK or KEK, that process shall be described in the TSS. h:br/>No explicit testing of the formation of the submask from the input password is required.*

*FCS_CKM_EXT.6.2: 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 (from the [AppPP]).*

***Guidance***
*Support for minimum length: The evaluators shall check the Operational Guidance to determine that there are instructions on how to generate large passwords/passphrases, and instructions on how to configure the password/passphrase length to provide entropy commensurate with the keys that the authorization factor is protecting.*

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

- ***Test 1:** Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.*
- ***Test 2:** Ensure that the TOE does not accept more than the maximum number of characters specified in FCS_CKM_EXT.6.1.*
- ***Test 3:** Ensure that the TOE does not accept less than the minimum number of characters specified in FCS_CKM_EXT.6.4. If the minimum length is settable by the administrator, the evaluator determines the minimum length or lengths to test.*
- ***Test 4:** Ensure that the TOE supports passwords consisting of all characters listed in FCS_CKM_EXT.6.2.*

*Conditioning: No explicit testing of the formation of the authorization factor from the input password/passphrase is required.*

**FCS_VAL_EXT.2(1) Validation Remediation (Server Administrator)**

***This is a selection-based component. Its inclusion depends upon selection from FIA_AUT_EXT.1.1.***

FCS_VAL_EXT.2.1(1)

The TSF shall **[selection:**

- ***institute a delay such that only [assignment: ST author specified number or configurable range of attempts] validation attempts can be made within a 24 hour period,***
- ***block validation after [assignment: ST author specified number or configurable range of attempts] of consecutive failed validation attempts***

**]**.

**Application Note:** This requirement must be claimed by the TOE if the ST author chooses "provide user authorization" in FIA_AUT_EXT.1.1.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine which remediation options are supported for which authentication options.*

***Guidance***
*If the remediation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established.*

***Tests***

*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. For each authentication factor supported, the evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any "lockout" period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.*

**FIA_CHR_EXT.1 Challenge/Response Recovery Credential**

***This is a selection-based component. Its inclusion depends upon selection from FIA_REC_EXT.1.1.***

FIA_CHR_EXT.1.1

The TSF shall generate a response only if it is able to access recovery information for [**selection**: *the user requesting the recovery, the user requesting recovery and the device for which the recovery was requested*].

**Application Note:** This requires that the TSF has the ability to attribute key chain information to the appropriate user(s).

FIA_CHR_EXT.1.2

The response shall work only for the user to whom it was generated.

**Application Note:** This mechanism is intended to provide a recovery method for a user who has forgotten their authentication factor and is unable to access their encrypted data on a system that is fully functional.

FIA_CHR_EXT.1.3

The response shall be used only during the same session in which the request was generated.

**Application Note:** The intent of this requirement is to limit the attack surface of the recovery credential mechanism by preventing the use of the credential following a reboot of the device.

FIA_CHR_EXT.1.4

The TSF shall generate an ephemeral response that has at least as many potential values as a corresponding password or PIN.

FIA_CHR_EXT.1.5

The TSF shall allow a maximum of [**assignment**: *integer value*] response entry attempts per boot cycle.

FIA_CHR_EXT.1.6

The TSF shall perform remediation as defined in FCS_VAL_EXT.2(2) for failed challenge recovery attempts.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that the methods for requesting a Recovery credential are specified. The TSS shall also describe the methods used to verify user requesting the Recovery credential. The evaluator shall also verify that the TSS contains the estimation of the strength of the ephemeral response and that it has at least as many potential values as a corresponding password or PIN.*

***Guidance***
*The evaluator shall confirm that the guidance documentation contains instructions for enforcing verification of the user for which the Recovery credential is requested. The guidance shall also describe configuring of the limit for consecutive failed validation attempts if this value is configurable.*

***Tests***
*The evaluator shall ensure that a response is only generated if the user for which recovery is requested are verified as specified in TSS. The evaluator shall also ensure that the response is applicable only on behalf of the requesting user with the constraints specified for consecutive failed authentication attempts.*

*The term “managed” below is used to refer a user or device which is registered on the server, i.e. their identity can be successfully verified by either administrator or TSF. The “unmanaged” presumes that the user/device cannot be successfully verified.*

*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall configure the Challenge/Response recovery to validate the user. The evaluator shall then issue a challenge on behalf of a managed user and ensure that TSF successfully generates the response.*
- ***Test 2:*** *The evaluator shall configure Challenge/Response recovery to validate the user. The evaluator shall then issue a challenge on behalf of managed User A and attempt to use it as an unmanaged User B to obtain a response. This should fail.*

- ***Test 3:*** *The evaluator shall issue a challenge on behalf of a managed user and ensure that the response received successfully will log the user in on that device.*
- ***Test 4:*** *The evaluator shall attempt to reuse the response of User A with User B on the same system and it should fail.*
- ***Test 5:*** *The evaluator shall issue a challenge on behalf of a managed user from a managed system, reboot the system [system terminates the session] and enter the response. This should fail.*
- ***Test 6:*** *The evaluator shall issue a challenge on behalf of a managed user and attempt to enter an incorrect response on the system the number of times described in the Guidance Documents. The observed behavior shall conform to the assignments/selections in FIA_CHR_EXT.1.5 and FIA_CHR_EXT.1.6.*

**FTP_TRP.1 Trusted Path**

***This is a selection-based component. Its inclusion depends upon selection from FMT_SMR.2.3.***

FTP_TRP.1.1

The TSF shall **be capable of using [selection: *IPsec as defined in the PP-Module for VPN Client, HTTPS in accordance with FCS_HTTPS_EXT.1 (from [AppPP]), TLS as defined in the Package for Transport Layer Security, SSH as defined in the Extended Package for Secure Shell*] to** provide a communication path between itself and **authorized remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [*modification, disclosure*].

FTP_TRP.1.2

The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for [*initial **administrator** authentication, [all remote administration actions]*].

**Application Note:** This SFR is intended to define protected communications between the Management Server and remote administrators.

**Evaluation Activity**

***TSS***
*The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

***Guidance***
*The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method are tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.*
- ***Test 2:*** *For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.*
- ***Test 3:*** *The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.*
- ***Test 4:*** *The evaluator shall ensure that, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.*

*Further evaluation activities are associated with the specific protocols.*

# Appendix C - Objective SFRs

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

This PP-Module does not define any objective SFRs.

# Appendix D - Extended Component Definitions

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

## D.1 Background and Scope

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

| Functional Class | Functional Components |
|---|---|
| Cryptographic Support (FCS) | FCS_CKM_EXT Cryptographic Key Management<br>FCS_IV_EXT Initialization Vector Generation<br>FCS_KDF_EXT Cryptographic Key Derivation Function<br>FCS_KYC_EXT Key Chaining and Key Storage<br>FCS_SMC_EXT Submask Combining<br>FCS_VAL_EXT Validation |
| Identification and Authentication (FIA) | FIA_AUT_EXT Authorization<br>FIA_REC_EXT Recovery Support |
| Protection of the TSF (FPT) | FPT_KYP_EXT Protection of Key and Key Material |
| Identification and Authentication (FIA) | FIA_CHR_EXT Challenge/Response Recovery Credential |

## D.2 Extended Component Definitions

### FCS_CKM_EXT Cryptographic Key Management

Components in this family define requirements for key management activities that are beyond the scope of what is defined in the FCS_CKM family in CC Part 2.

### Component Leveling

FCS_CKM_EXT.4, Cryptographic Key Destruction, describes supported methods for key destruction.

### Management: FCS_CKM_EXT.4

The following actions could be considered for the management functions in FMT:

- Manually perform cryptographic erasure.

### Audit: FCS_CKM_EXT.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Basic: Manual erasure of cryptographic data.

### FCS_CKM_EXT.4 Cryptographic Key Destruction

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FCS_CKM_EXT.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**selection**:

- *For volatile memory, the destruction shall be executed by a [**selection**:*
  - *single overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's RBG, zeroes, ones, new value of a key, [**assignment**: any value that does not contain any CSP]] ,*
  - *removal of power to the memory,*
  - *destruction of reference to the key directly followed by a request for garbage collection*

  *] ,*
- *For non-volatile memory, the destruction shall be executed by [**selection**:*
  - *destruction of all KEKs protecting the target key, where none of the KEKs protecting the target key are derived ,*
  - *the invocation of an interface provided by the underlying platform that [**selection**:*
    - *logically addresses the storage location of the key and performs a [**selection**: single, [**assignment**: ST author defined multi-pass]] overwrite consisting of [**selection**: a pseudo-random pattern using the TSF's RBG, zeroes, ones, new value of a key, [**assignment**: any value that does not contain any CSP]] ,*
    - *instructs the underlying platform to destroy the abstraction that represents the key*

    *]*

*]*

].

### FCS_CKM_EXT.4.2

The TSF shall destroy all keys and key material when no longer needed.

### Component Leveling

FCS_CKM_EXT.6, Cryptographic Password/Passphrase Conditioning, requires the TSF to implement password/passphrase conditioning using a specified algorithm and with specific constraints on the password/passphrase composition.

### Management: FCS_CKM_EXT.6

There are no specific management functions identified.

### Audit: FCS_CKM_EXT.6

There are no auditable events foreseen.

### FCS_CKM_EXT.6 Cryptographic Password/Passphrase Conditioning

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation
FCS_RBG_EXT.1 Random Bit Generation Services

### FCS_CKM_EXT.6.1

The TSF shall support a password/passphrase of up to [**assignment**: *maximum password size, positive integer of 64 or more*] characters used to generate a password authorization factor.

### FCS_CKM_EXT.6.2

The TSF shall allow passwords 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*].

### FCS_CKM_EXT.6.3

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*] bits that meet the following: [*NIST SP 800-132*].

### FCS_CKM_EXT.6.4

The TSF shall not accept passwords less than [**selection**: *a value settable by the administrator*, *[**assignment**: minimum password length accepted by the TOE, must be >= 1]*] and greater than the maximum password length defined in FCS_CKM_EXT.6.1.

### FCS_CKM_EXT.6.5

The TSF shall generate all salts using an RBG that meets FCS_RBG_EXT.1 **(from [AppPP])** and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM_EXT.6.3.

### FCS_IV_EXT Initialization Vector Generation

Components in this family define requirements for initialization vector generation.

### Component Leveling

FCS_IV_EXT.1, Initialization Vector Generation, specifies the required initialization vector generation methods used by the TSF for various cryptographic algorithms.

### Management: FCS_IV_EXT.1

There are no specific management functions identified.

### Audit: FCS_IV_EXT.1

There are no auditable events foreseen.

### FCS_IV_EXT.1 Initialization Vector Generation

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_IV_EXT.1.1

The TSF shall [**selection**:

- *invoke platform-provided functionality to generate IVs,*
- *generate IVs with the following properties [**selection**:*
  - *CBC: IVs shall be non-repeating and unpredictable,*
  - *CCM: Nonce shall be non-repeating and unpredictable,*
  - *XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,*
  - *GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key*

  *]*

].

### FCS_KDF_EXT Cryptographic Key Derivation Function

Components in this family define requirements for the implementation of cryptographic key derivation functions

### Component Leveling

FCS_KDF_EXT.1, Cryptographic Key Derivation Function, requires the TSF to specify how it performs key derivation.

### Management: FCS_KDF_EXT.1

The following actions could be considered for the management functions in FMT:

- Configuration of the cryptographic functionality.

### Audit: FCS_KDF_EXT.1

There are no auditable events foreseen.

### FCS_KDF_EXT.1 Cryptographic Key Derivation Function

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation
FCS_RBG_EXT.1 Random Bit Generation Services

### FCS_KDF_EXT.1.1

The TSF shall [**selection**:

- *not derive keys*,
- *accept [**selection**: a submask generated by an RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**, a conditioned password, an imported submask] to derive an intermediate key, as defined in [**selection**:*
  - *NIST SP 800-108 [**selection**: KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iteration Mode] ,*
  - *NIST SP 800-132*

  *] using the keyed-hash functions specified in FCS_COP.1**(4) (from [AppPP])**, such that the output is at least of equivalent security strength (in number of bits) to the [FEK(s)]*

].

### FCS_KYC_EXT Key Chaining and Key Storage

Components in this family define requirements for the secure storage of keys through the use of a logical key chain.

### Component Leveling

FCS_KYC_EXT.1, Key Chaining and Key Storage, requires the TSF to specify how it implements key chaining.

### Management: FCS_KYC_EXT.1

The following actions could be considered for the management functions in FMT:

- Configuration of the cryptographic functionality.

### Audit: FCS_KYC_EXT.1

There are no auditable events foreseen.

### FCS_KYC_EXT.1 Key Chaining and Key Storage

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation
FCS_KDF_EXT.1 Cryptographic Key Derivation Function
FCS_SMC_EXT.1 Submask Combining

### FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of [*[intermediate keys] originating from **one or more initial [selection: submask(s), recovery value(s)]** to [the final value returned to the endpoint] using the following method(s): [**selection**:*

- *utilization of the platform key storage,*
- *utilization of platform key storage that performs key wrap with a TSF provided key,*
- *implementation of key derivation as specified in FCS_KDF_EXT.1,*
- *implementation of key wrapping as specified in FCS_COP.1**(5)**,*
- *implementation of key combining as specified in FCS_SMC_EXT.1,*
- *implementation of key encryption as specified in FCS_COP.1**(7)**,*
- *implementation of key transport as specified in FCS_COP.1**(6)***

*] while maintaining an effective strength of [**selection**:*

- *[**selection**: 128 bits, 256 bits] for symmetric keys ,*
- *[**selection**: 128 bits, 192 bits, 256 bits] for asymmetric keys*

*]* commensurate with the strength of the FEK and [**selection**:

- *no supplemental key chains*,
- *other supplemental key chains that protect a key or keys in the primary key chain using the following method(s): [**selection**:*
  - *utilization of the platform key storage,*
  - *utilization of platform key storage that performs key wrap with a TSF provided key,*
  - *implementation of key wrapping as specified in FCS_COP.1**(5)**,*
  - *implementation of key combining as specified in FCS_SMC_EXT.1,*
  - *implementation of key encryption as specified in FCS_COP.1**(7)**,*
  - *implementation of key transport as specified in FCS_COP.1**(6)**,*
  - *implementation of key derivation as specified in FCS_KDF_EXT.1*

  *]*

].

### FCS_SMC_EXT Submask Combining

Components in this family define requirements for generation of intermediate keys via submask combining.

### Component Leveling

FCS_SMC_EXT.1, Submask Combining , requires the TSF to implement submask combining in a specific manner to support the generation of intermediate keys.

### Management: FCS_SMC_EXT.1

The following actions could be considered for the management functions in FMT:

- Configuration of the cryptographic functionality.

### Audit: FCS_SMC_EXT.1

There are no auditable events foreseen.

### FCS_SMC_EXT.1 Submask Combining

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_SMC_EXT.1.1

The TSF shall [**selection**:

- *not perform submask combining*,
- *combine submasks using the following method [**selection**: exclusive OR (XOR), SHA-256, SHA-384, SHA-512, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] to generate an intermediate key*

].

### FCS_VAL_EXT Validation

Components in this family define requirements for validation of data supplied to the TOE and any consequences resulting from failed validation attempts.

### Component Leveling

FCS_VAL_EXT.1(1), Validation (Server Administrator), requires the TSF to specify what data is being validated and how the validation is performed.

### Management: FCS_VAL_EXT.1(1)

There are no specific management functions identified.

### Audit: FCS_VAL_EXT.1(1)

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Change to configuration of validation function behavior.

### FCS_VAL_EXT.1(1) Validation (Server Administrator)

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_VAL_EXT.1.1(1)

The TSF shall perform validation of the [*admin*] by [**selection**:

- *receiving assertion of the subject's validity from [**assignment**: Operational Environment component responsible for authentication]*,
- *validating the [**selection**: submask, intermediate key] using the following methods: [**selection**:*
  - *key wrap as specified in FCS_COP.1**(5)***,
  - *hash the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(2) (from [AppPP])** and compare it to a stored hash*,
  - *decrypt a known value using the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(1) (from [AppPP])** and compare it against a stored known value*

  *]*

].

### FCS_VAL_EXT.1.2(1)

The TSF shall require validation of the [*admin*] prior to [*permitting the actions described in FMT_MTD.1.1 and FMT_SMF.1.1(2)*].

### Component Leveling

FCS_VAL_EXT.1(2), Validation (User),

### Management: FCS_VAL_EXT.1(2)

There are no management functions foreseen.

### Audit: FCS_VAL_EXT.1(2)

There are no audit events foreseen.

### FCS_VAL_EXT.1(2) Validation (User)

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FCS_VAL_EXT.1.1(2)

The TSF shall perform validation of the [*user*] by [**selection**:

- *receiving assertion of the subject's validity from [**assignment**: Operational Environment component responsible for authentication]*,
- *validating the [**selection**: submask, intermediate key] using the following methods: [**selection**:*
  - *key wrap as specified in FCS_COP.1**(5)***,
  - *hash the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(2) (from***

***[AppPP])*** *and compare it to a stored hash,*
- *decrypt a known value using the [**selection**: submask, intermediate key, FEK] as specified in FCS_COP.1**(1) (from [AppPP])** and compare it against a stored known value*

*]*

].

### FCS_VAL_EXT.1.2(2)

The TSF shall require validation of the [*user*] prior to [*transmitting submasks, FEKs, or keys to decrypt FEKs to the endpoint*].

### Component Leveling

FCS_VAL_EXT.2(2), Validation Remediation (User),

### Management: FCS_VAL_EXT.2(2)

There are no management functions foreseen.

### Audit: FCS_VAL_EXT.2(2)

There are no audit events foreseen.

### FCS_VAL_EXT.2(2) Validation Remediation (User)

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FCS_VAL_EXT.2.1(2)

The TSF shall [**selection**:

- *[[**selection**: direct the endpoint to perform key sanitization, perform key sanitization] of FEK(s) or an intermediate key] upon **[assignment: ST specified number or configurable range of]** consecutive failed validation attempts,*
- *institute a delay such that only [**assignment**: ST author specified number **or configurable range** of attempts] can be made within a 24 hour period,*
- *block validation after [**assignment**: ST author specified number **or configurable range** of attempts] of consecutive failed validation attempts,*
- ***terminate the session** after [**assignment**: ST author specified number **or configurable range** of attempts] of consecutive failed validation attempts*

].

### Component Leveling

FCS_VAL_EXT.2(1), Validation Remediation (Server Administrator), requires the TSF to specify what the TOE's response is in the event of a data validation failure.

### Management: FCS_VAL_EXT.2(1)

The following actions could be considered for the management functions in FMT:

- Configuration of the number of failed validation attempts required to trigger corrective behavior.
- Configuration of the corrective behavior to issue in the event of an excessive number of failed validation attempts.

### Audit: FCS_VAL_EXT.2(1)

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Triggering of excessive validation failure response behavior.

### FCS_VAL_EXT.2(1) Validation Remediation (Server Administrator)

Hierarchical to: No other components.

Dependencies to: FCS_VAL_EXT.1 Validation

### FCS_VAL_EXT.2.1(1)

The TSF shall **[selection:**

- ***institute a delay such that only [assignment: ST author specified number or configurable range of attempts] validation attempts can be made within a 24 hour period,***
- ***block validation after [assignment: ST author specified number or configurable range of attempts] of consecutive failed validation attempts***

**]**.

### FIA_AUT_EXT Authorization

Components in this family define requirements for how subject authorization is performed. Where FIA_UAU in CC Part 2 defines circumstances where authentication is required, this family describes the specific computational methods used to determine whether a subject's presented authentication data is valid.

### Component Leveling

FIA_AUT_EXT.1, Subject Authorization, specifies the manner in which the TSF performs user authorization.

### Management: FIA_AUT_EXT.1

The following actions could be considered for the management functions in FMT:

- Configuration of authentication factors.

### Audit: FIA_AUT_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Failure of authorization function.
- Basic: All use of authorization function.

### FIA_AUT_EXT.1 Subject Authorization

Hierarchical to: No other components.

Dependencies to: FCS_CKM_EXT.6 Cryptographic Password/Passphrase Conditioning
FCS_RBG_EXT.1 Random Bit Generation Services

### FIA_AUT_EXT.1.1

The TSF shall [**selection**: *receive assertion of the user's validity from: [**assignment**: Operational Environment component responsible for user authentication], provide authorization*] based on [**selection**:

- *a password authorization factor conditioned as defined in FCS_CKM_EXT.6,*
- *an external smart card factor that is at least the same bit-length as the FEK(s), and is protecting a submask that is [**selection**: generated by the TOE (using the RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**), generated by the platform] protected using RSA with key size [**selection**: 3072 bits, 4096 bits] with user presence proved by presentation of the smart card and [**selection**: no PIN, an OE defined PIN, a configurable PIN] ,*
- *an external USB token factor that is at least the same security strength as the FEK(s), and is providing a submask generated by the [**selection**: TOE, using the RBG as specified in FCS_RBG_EXT.1 **(from [AppPP])**, platform]*

].

### FIA_REC_EXT Recovery Support

Components in this family define the TOE's support for recovery credentials as an alternate method for user authorization.

### Component Leveling

FIA_REC_EXT.1, Recovery Support, requires the TSF to specify the supported recovery method and to include a means to enable/disable any supported recovery method.

### Management: FIA_REC_EXT.1

The following actions could be considered for the management functions in FMT:

- Ability to enable/disable the user of recovery credentials.
- Ability to change recovery credential values.

### Audit: FIA_REC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Configuration of recovery methods.

### FIA_REC_EXT.1 Recovery Support

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FIA_REC_EXT.1.1

The TSF shall [**selection**: *provide the ability to enable and disable the use of recovery credentials, not support recovery*].

### FIA_REC_EXT.1.2

The TSF shall support the following recovery mechanisms [**selection**: *Challenge Response Recovery as defined in FIA_CHR_EXT.1, None*].

### FPT_KYP_EXT Protection of Key and Key Material

Components in this family define requirements for secure storage of keys.

### Component Leveling

FPT_KYP_EXT.1, Protection of Keys and Key Material , requires the TSF to protect stored key data in a specified manner.

### Management: FPT_KYP_EXT.1

The following actions could be considered for the management functions in FMT:

- Configuration of the cryptographic functionality.

### Audit: FPT_KYP_EXT.1

There are no auditable events foreseen.

### FPT_KYP_EXT.1 Protection of Keys and Key Material

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation
FCS_KDF_EXT.1 Cryptographic Key Derivation Function
FCS_KYC_EXT.1 Key Chaining and Key Storage
FCS_SMC_EXT.1 Submask Combining
FCS_STO_EXT.1 Storage of Credentials

### FPT_KYP_EXT.1.1

The TSF shall store keys in non-volatile memory only when [**selection**:

- *wrapped, as specified in FCS_COP.1(5),*

- *encrypted, as specified in FCS_COP.1(1)* ***(from [AppPP])***,
- *the plaintext key is stored in the underlying platform's keystore as specified by FCS_STO_EXT.1.1* ***(from [AppPP])***,
- *the plaintext key is stored in a SQL database in the Operational Environment,*
- *the plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1.,*
- *the plaintext key will no longer provide access to the encrypted data after initial provisioning,*
- *the plaintext key is a key split that is combined as specified in FCS_SMC_EXT.1 and another contribution to the split is [**selection**: wrapped as specified in FCS_COP.1(5) or encrypted as specified in FCS_COP.1(7), derived and not stored in non-volatile memory] ,*
- *the plaintext key is stored on an external storage device for use as an authorization factor.,*
- *the plaintext key is used to encrypt a key as specified in FCS_COP.1(7) or wrap a key as specified in FCS_COP.1(5) that is already encrypted as specified in FCS_COP.1(7) or wrapped as specified in FCS_COP.1(5)*

].

### Component Leveling

FPT_KYP_EXT.2, Attribution of Key and Key Material, requires the TSF to protect stored key data in a specified manner.

### Management: FPT_KYP_EXT.2

The following actions could be considered for the management functions in FMT:

- Registration of users.
- Revocation of user registration.

### Audit: FPT_KYP_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Basic: Creation and revocation of user registration.

### FPT_KYP_EXT.2 Attribution of Key and Key Material

Hierarchical to: No other components.

Dependencies to: FPT_ITT.1 Basic Internal TSF Data Transfer Protection

#### FPT_KYP_EXT.2.1

The TSF shall maintain an association between stored endpoint keys and user identity, [**selection**: *remote endpoints*, *recovery credential*, *system identity*, *no other subjects*].

#### FPT_KYP_EXT.2.2

The TSF shall provide the ability to register users by exchange of [**assignment**: *mutually identifying information that allows for an association to be made*].

#### FPT_KYP_EXT.2.3

The TSF shall provide the ability to revoke the registration of users by [**assignment**: *method of removing and/or exchanging information that prevents further communications between the TOE and the endpoint*].

#### FPT_KYP_EXT.2.4

The TSF shall transmit any secure or private cryptographic information that is transferred between the TOE and a user's endpoint in order to establish or disestablish an association using a communications channel with a security strength at least as great as the strength of the information being transmitted.

### FIA_CHR_EXT Challenge/Response Recovery Credential

Components in this family define requirements for the use of challenge/response as a recovery method.

### Component Leveling

FIA_CHR_EXT.1, Challenge/Response Recovery Credential, requires the TSF to implement a challenge/response method to generate recovery credentials for an authorized user.

### Management: FIA_CHR_EXT.1

The following actions could be considered for the management functions in FMT:

- Ability to enable/disable the user of recovery credentials.
- Ability to change recovery credential values.

### Audit: FIA_CHR_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Failure of recovery attempt.
- Basic: All recovery attempts.

### FIA_CHR_EXT.1 Challenge/Response Recovery Credential

Hierarchical to: No other components.

Dependencies to: FCS_VAL_EXT.1 Validation
FIA_REC_EXT.1 Recovery Support

#### FIA_CHR_EXT.1.1

The TSF shall generate a response only if it is able to access recovery information for [**selection**: *the user requesting the recovery*, *the user requesting recovery and the device for which the recovery was requested*].

#### FIA_CHR_EXT.1.2

The response shall work only for the user to whom it was generated.

### FIA_CHR_EXT.1.3

The response shall be used only during the same session in which the request was generated.

### FIA_CHR_EXT.1.4

The TSF shall generate an ephemeral response that has at least as many potential values as a corresponding password or PIN.

### FIA_CHR_EXT.1.5

The TSF shall allow a maximum of [**assignment**: *integer value*] response entry attempts per boot cycle.

### FIA_CHR_EXT.1.6

The TSF shall perform remediation as defined in FCS_VAL_EXT.2(2) for failed challenge recovery attempts.

# Appendix E - Key Management Description

This appendix should be combined with the appendix in the File Encryption Module if it is also being evaluated. The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

Essay:

The essay will provide the following information for all keys in the key chain:

- The purpose of the key
- If the key is stored in non-volatile memory
- How and when the key is protected
- How and when the key is derived
- The strength of the key
- When or if the key would be no longer needed, along with a justification
- How and when a key may be transmitted

The essay will also describe the following topics:

- A description of all authorization factors that are supported by the product and how each factor is handled, including any conditioning and combining performed.
- If validation is implemented, the process for validation shall be described, noting what value is used for validation and the process used to perform the validation. It shall describe how this process ensures no keys in the key chain are weakened or exposed by this process.
- The authorization process that leads to the recovery or access by an end user or administrator. This section shall detail the key chain used by the product. It shall describe which keys are used in the protection of the FEK(s) or KEK(s) and how they meet the encryption or derivation requirements, including the direct chain from the initial authorization to the FEK(s) or KEK(s). It shall also include any values that add into that key chain or interact with the key chain and the protections that ensure those values do not weaken or expose the overall strength of the key chain.
- The diagram and essay will clearly illustrate the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or all of the initial authorization values and the effective strength of the FEK(s) is maintained throughout the key chain.
- A description of the data encryption engine, its components, and details about its implementation (e.g. initialization of the product, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and how resources to be encrypted are identified. The description should also include the data flow from the device's host interface to the device's persistent media storing the data or transmission to and endpoint, information on those conditions in which the data bypasses the data encryption engine. The description should be detailed enough to verify all platforms ensure that when the user enables encryption, the product encrypts all selected resources.
- The process for destroying keys when they are no longer needed by describing the storage location of all keys and the protection of all keys stored in non-volatile memory.

Diagram:

- The diagram will include all keys from the initial authorization factor(s) to the FEK(s) and any keys or values that contribute into the chain. It must list the cryptographic strength of each key and indicate how each key along the chain is protected with either options from key chaining requirement. The diagram should indicate the input used to derive or decrypt each key in the chain.
- A functional (block) diagram showing the main components (such as memories and processors) the initial steps needed for the activities the TOE performs to ensure it encrypts the targeted resources when a user or administrator first provisions the product.

# Appendix F - Bibliography


| Identifier | Title |
| --- | --- |
| [CC] | Common Criteria for Information Technology Security Evaluation -<br>• Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.<br>• Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.<br>• Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017. |
| [AppPP] | Protection Profile for Application Software, Version 1.3 |
| [FIPS140-2] | Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology, March 19, 2007 |
| [FIPS180-4] | Federal Information Processing Standards Publication (FIPS-PUB) 180-4, Secure Hash Standard, March, 2012 |
| [FIPS186-4] | Federal Information Processing Standard Publication (FIPS-PUB) 186-4, Digital Signature Standard (DSS), National Institute of Standards and Technology, July 2013 |
| [FIPS197] | Federal Information Processing Standards Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001 |
| [FIPS198-1] | Federal Information Processing Standards Publication (FIPS-PUB) 198-1, The Keyed-Hash Message Authentication Code (HMAC), July 2008 |
| [NIST800-38A] | NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, 2001 Edition |
| [NIST800-56A] | NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised), March 2007 |
| [NIST800-56B] | NIST Special Publication 800-56B, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography, August 2009 |
| [NIST800-90] | NIST Special Publication 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised), March 2007 |
| [NIST800-132] | NIST Special Publication 800-132, Recommendation for Password-Based Key Derivation, December 2010 |
| [NIST800-38F] | NIST Special Publication 800-38F,Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, December 2012 |

# Appendix G - Acronyms

| Acronym | Meaning |
|---|---|
| AES | Advanced Encryption Standard |
| CC | Common Criteria |
| FAK | File Authentication Key |
| FEK | File Encryption Key |
| DRBG | Deterministic Random Bit Generator |
| EAL | Evaluation Assurance Level |
| ECC | Elliptic Curve Cryptography |
| ECC CDH | Elliptic Curve Cryptography Cofactor Diffie-Hellman (see NIST SP 800-56A rev 2, section 6.2.2.2) |
| FIPS | Federal Information Processing Standards |
| ISSE | Information System Security Engineers |
| IT | Information Technology |
| KDF | Key Derivation Function |
| KEK | Key Encryption Key |
| PBKDF | Password-Based Key Derivation Function |
| PIN | Personnel Identification Number |
| PKI | Public Key Infrastructure |
| PP | Protection Profile |
| PUB | Publication |
| RBG | Random Bit Generator |
| SAR | Security Assurance Requirement |
| SF | Security Function |
| SFR | Security Functional Requirement |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |