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 Source	This cryptographic function provides a seed for a random bit generator by accumulating the outputs from one or more noise sources. The functionality includes a measure of the minimum work required to guess a given output and tests to ensure that the noise sources are operating properly.
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_S is countered by O.KEY_MATERIAL_SERV this provides for proper protection of key m on the server.
T.MAN_IN_THE_MIDDLE	O.SECURE_CHANNEL	The threat T.MAN_IN_THE_MIDDLE is co by O.SECURE_CHANNEL as this protects man in the middles attacks.
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	O.VERIFIED_ADMIN	The threat T.UNAUTHORIZED_ADMINISTRATOR_A is countered by O.VERIFIED_ADMIN as th provides methods to verify the administrat

T.UNTRUSTED_COMMUNICATION_CHANNELS	O.SECURE_CHANNEL	The threat T.UNTRUSTED_COMMUNICATION_CHA is countered by O.SECURE_CHANNEL as provides a trusted channel for any server e communications.
T.UNAUTHORIZED_DATA_ACCESS_ENDPOINT	O.ENTERPRISE_KEY_PROTECTION	The threat T.UNAUTHORIZED_DATA_ACCESS_EN is countered by O.ENTERPRISE_KEY_PROTECTION as provides for encryption of keys that protec
T.UNAUTHORIZED_DATA_ACCESS_SERVER	O.RECOVERY_PROTECTION	The threat T.UNAUTHORIZED_DATA_ACCESS_SE countered by O.RECOVERY_PROTECTI(this provides for protection of recovery infc
A.ENVIRONMENTAL_STORAGE	OE.ENVIRONMENTAL_STORAGE	The operational environment objective OE.ENVIRONMENTAL_STORAGE is real through A.ENVIRONMENTAL_STORAGE
A.PHYSICAL_SERVER	OE.PHYSICAL_SERVER	The operational environment objective OE.PHYSICAL_SERVER is realized throu A.PHYSICAL_SERVER.
A.SECURED_CONFIGURATION	OE.SECURED_CONFIGURATION	The operational environment objective OE.SECURED_CONFIGURATION is reali through A.SECURED_CONFIGURATION.
A.SECURED_ENVIRONMENT	OE.SECURED_ENVIRONMENT	The operational environment objective OE.SECURED_ENVIRONMENT is realize 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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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*

].

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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 FCS_IV_EXT Initialization Vector Generation FCS_KDF_EXT Cryptographic Key Derivation Function FCS_KYC_EXT Key Chaining and Key Storage FCS_SMC_EXT Submask Combining FCS_VAL_EXT Validation
Identification and Authentication (FIA)	FIA_AUT_EXT Authorization 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 - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
[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