Supporting Document
Mandatory Technical Document

PP-Module for File Encryption Enterprise Management

Version: 1.0

2019-07-30

National Information Assurance Partnership

Foreword

This is a Supporting Document (SD), intended to complement the Common Criteria version 3 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

SDs may be “Guidance Documents”, that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or “Mandatory Technical Documents”, whose application is mandatory for evaluations whose scope is covered by that of the SD. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

Technical Editor:
National Information Assurance Partnership (NIAP)

Document history:

Version	Date	Comment
1.0	2019-07-30	Initial Release

General Purpose:
The purpose of this SD is to define evaluation methods for the functional behavior of file encryption enterprise management products.

Acknowledgements:
This SD was developed with support from NIAP File Encryption Enterprise Management Technical Community members, with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

Table of Contents

1 Introduction

1.1 Technology Area and Scope of Supporting Document

The scope of the File Encryption Enterprise Management PP-Module is to describe the security functionality of a file encryption enterprise management in terms of [CC] and to define functional and assurance requirements for them. The PP-Module is intended for use with the Application Software Protection Profile.

This SD is mandatory for evaluations of TOEs that claim conformance to a PP-Configuration that includes the PP-Module for File Encryption Enterprise Management, Version 1.0. Although Evaluation Activities are defined mainly for the evaluators to follow, in general they also help Developers to prepare for evaluation by identifying specific requirements for their TOE. The specific requirements in Evaluation Activities may in some cases clarify the meaning of Security Functional Requirements (SFR), and may identify particular requirements for the content of Security Targets (ST) (especially the TOE Summary Specification), user guidance documentation, and possibly supplementary information (e.g. for entropy analysis or cryptographic key management architecture).

1.2 Structure of the Document

Evaluation Activities can be defined for both SFRs and Security Assurance Requirements (SAR), which are themselves defined in separate sections of the SD.

If any Evaluation Activity cannot be successfully completed in an evaluation then the overall verdict for the evaluation is a 'fail'. In rare cases there may be acceptable reasons why an Evaluation Activity may be modified or deemed not applicable for a particular TOE, but this must be approved by the Certification Body for the evaluation.

In general, if all Evaluation Activities (for both SFRs and SARs) are successfully completed in an evaluation then it would be expected that the overall verdict for the evaluation is a ‘pass’. To reach a ‘fail’ verdict when the Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

Similarly, at the more granular level of Assurance Components, if the Evaluation Activities for an Assurance Component and all of its related SFR Evaluation Activities are successfully completed in an evaluation then it would be expected that the verdict for the Assurance Component is a ‘pass’. To reach a ‘fail’ verdict for the Assurance Component when these Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

1.3 Terms

Common Criteria Terms

The following definitions are for Common Criteria terms used in this document:

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.

Technical Terms

The following definitions define Technical terms used in this document:

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.

2 Evaluation Activities for SFRs

The EAs presented in this section capture the actions the evaluator performs to address technology specific aspects covering specific SARs (e.g. ASE_TSS.1, ADV_FSP.1, AGD_OPE.1, and ATE_IND.1) – this is in addition to the CEM work units that are performed in 6 Evaluation Activities for SARs.

Regarding design descriptions (designated by the subsections labelled TSS, as well as any required supplementary material that may be treated as proprietary), the evaluator must ensure there is specific information that satisfies the EA. For findings regarding the TSS section, the evaluator's verdicts will be associated with the CEM work unit ASE_TSS.1-1. Evaluator verdicts associated with the supplementary evidence will also be associated with ASE_TSS.1-1, since the requirement to provide such evidence is specified in ASE in the cPP.

For ensuring the guidance documentation provides sufficient information for the administrators/users as it pertains to SFRs, the evaluator's verdicts will be associated with CEM work units ADV_FSP.1-7, AGD_OPE.1-4, and AGD_OPE.1-5.

Finally, the subsection labelled Tests is where the authors have determined that testing of the product in the context of the associated SFR is necessary. While the evaluator is expected to develop tests, there may be instances where it is more practical for the developer to construct tests, or where the developer may have existing tests. Therefore, it is acceptable for the evaluator to witness developer-generated tests in lieu of executing the tests. In this case, the evaluator must ensure the developer's tests are executing both in the manner declared by the developer and as mandated by the EA. The CEM work units that are associated with the EAs specified in this section are: ATE_IND.1-3, ATE_IND.1-4, ATE_IND.1-5, ATE_IND.1-6, and ATE_IND.1-7.

2.1 Application Software Protection Profile

The EAs defined in this section are only applicable in cases where the TOE claims conformance to a PP-Configuration that includes the App PP.

2.1.1 Modified SFRs

2.1.1.1 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

TSS
For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality.

Guidance
None.

Tests
The evaluator shall perform the following tests.

- **Test 1:** The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, or SSH in accordance with the selection in the ST.
- **Test 2:** The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.
- **Test 3:** The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found.
  For iOS: If the platformIf "encrypt all transmitted data" is selected, the evaluator shall ensure that the application's Info.plist file does not contain the NSAllowsArbitraryLoads or NSExceptionAllowsInsecureHTTPLoads keys, as these keys disable iOS's Application Transport Security feature.

2.1.2 TOE SFR Evaluation Activities

2.1.3 Cryptographic Support (FCS)

FCS_CKM_EXT.4 Cryptographic Key Destruction

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

KMD

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

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

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

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

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

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

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

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

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

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

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

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

When the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, to mitigate this the drive should support the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.

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

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

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

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

  The evaluator shall:

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

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

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

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

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

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

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

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

KMD

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Variable Associated Data Test

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

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

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

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

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

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

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

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

Guidance
None.

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

- **Test 1:** The evaluator shall inspect each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE and make sure it is the correct length, either 256 or 384 bytes depending on RSA key size. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts that are the wrong length and verify that the erroneous input is detected and that the decryption operation exits with an error code.
- **Test 2:** The evaluator shall convert each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE to the correct cipher text integer, c, and use the decryption primitive to compute em = RSADP((n,d),c) and convert em to the encoded message EM. The evaluator shall then check that the first byte of EM is 0x00. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts where the first byte of EM was set to a value other than 0x00, and verify that the erroneous input is detected and that the decryption operation exits with an error code.
- **Test 3:** The evaluator shall decrypt each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE using RSADP, and perform the OAEP decoding operation (described in NIST SP 800-56B section 7.2.2.4) to recover HA' || X. For each HA', the evaluator shall take the corresponding A and the specified hash algorithm and verify that HA' = Hash(A). The evaluator shall also force the TOE to perform some RSA-OAEP decryption where the A value is passed incorrectly, and the evaluator shall verify that an error is detected.
- **Test 4:** The evaluator shall check the format of the 'X' string recovered in OAEP.Test.3 to ensure that the format is of the form PS || 01 || K, where PS consists of zero or more consecutive 0x00 bytes and K is the transported keying material. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts for which the resulting 'X' strings do not have the correct format (i.e., the leftmost non-zero byte is not 0x01). These incorrectly formatted 'X' variables shall be detected by the

RSA-OEAP decrypt function.
- **Test 5:** The evaluator shall trigger all detectable decryption errors and validate that the returned error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations are revealed to the sender.

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

- **Test 1:** The evaluator shall inspect each cipher text, C, produced by KTS-KEM-KWS encryption operation of the TOE and make sure the length (in bytes) of the cipher text, cLen, is greater than nLen (the length, in bytes, of the modulus of the RSA public key) and that cLen - nLen is consistent with the byte lengths supported by the key wrapping algorithm. The evaluator shall feed into the KTS-KEM-KWS decryption operation a cipher text of unsupported length and verify that an error is detected and that the decryption process stops.
- **Test 2:** The evaluator shall separate the cipher text, C, produced by the sender-side of the TOE into its C0 and C1 components and use the RSA decryption primitive to recover the secret value, Z, from C0. The evaluator shall check that the unsigned integer represented by Z is greater than 1 and less than n-1, where n is the modulus of the RSA public key. The evaluator shall construct examples where the cipher text is created with a secret value Z = 1 and make sure the KTS-KEM-KWS decryption process detects the error. Similarly, the evaluator shall construct examples where the cipher text is created with a secret value Z = n - 1 and make sure the KTS-KEM-KWS decryption process detects the error.
- **Test 3:** The evaluator shall attempt to successfully recover the secret value Z, derive the key wrapping key, KWK, and unwrap the KWA-cipher text following the KTS-KEM-KWS decryption process given in NISP SP 800-56B section 7.2.3.4. If the key-wrapping algorithm is AES-CCM, the evaluator shall verify that the value of any (unwrapped) associated data, A, that was passed with the wrapped keying material is correct The evaluator shall feed into the TOE's KTS-KEM-KWS decryption operation examples of incorrect cipher text and verify that a decryption error is detected. If the key-wrapping algorithm is AES-CCM, the evaluator shall attempt at least one decryption where the wrong value of A is given to the KTS-KEM-KWS decryption operation and verify that a decryption error is detected. Similarly, if the key-wrapping algorithm is AES-CCM, the evaluator shall attempt at least one decryption where the wrong nonce is given to the KTS-KEM-KWS decryption operation and verify that a decryption error is detected.
- **Test 4:** The evaluator shall trigger all detectable decryption errors and validate that the resulting error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations (in particular, no Z values) are revealed to the sender.

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

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

Guidance
None.

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

FCS_IV_EXT.1 Initialization Vector Generation

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

Guidance
None.

Tests
None.

FCS_KDF_EXT.1 Cryptographic Key Derivation Function

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

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

KMD

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

Guidance
None.

Tests
None.

FCS_KYC_EXT.1 Key Chaining and Key Storage

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

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

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

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

Tests
None.

FCS_SMC_EXT.1 Submask Combining

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

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

Guidance
None.

Tests
None.

FCS_VAL_EXT.1(1) Validation (Server Administrator)

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

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

validation takes place).

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

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

Tests
There are no test activities for this requirement.

FCS_VAL_EXT.1(2) Validation (User)

TSS
The evaluator shall examine the TSS to determine which component of the Operational Environment is used to assert the User’s identity.

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

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

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

Tests
The evaluator shall perform the following tests:

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

FCS_VAL_EXT.2(2) Validation Remediation (User)

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

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

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

2.1.4 Identification and Authentication (FIA)

FIA_AUT_EXT.1 Subject Authorization

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

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

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

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

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

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

FIA_REC_EXT.1 Recovery Support

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

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

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

FIA_UAU.1 Timing of Authentication

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

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

Tests
The evaluator shall perform the following tests:

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

FIA_UID.1 Timing of Identification

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

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

Tests
The evaluator shall perform the following tests:

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

2.1.5 Security Management (FMT)

FMT_MOF.1 Server Management of Security Functions Behavior

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

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

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

FMT_MTD.1 Management of TSF Data

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

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

Tests
The evaluator shall perform the following tests:

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

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

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

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

- Register new user
- Revoke registration of an user
- Initiate key generation
- Initiate key escrow

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

Tests

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

- **Test 1:** The evaluator shall configure the management server and two users according to the guidance documents. The evaluator shall register the users with the management server. The evaluator shall verify that the users are identified by the management server as defined in the guidance documents. This test shall pass.
- **Test 2:** The evaluator shall disconnect the second user from the network. The evaluator shall revoke the registration of the second user in the management server. The evaluator shall attempt to connect the second user to the network and verify the endpoint fails to connect or is displayed as revoked in the console.
- **Test 3:** The evaluator shall verify that the TOE performs the actions (e.g. generate key) and sends the result to the user's client. The user's client shall perform the actions necessary to accept the updated configuration (e.g. encrypt the data with the new key, update the encryption algorithm key size or mode and re-encrypt).
- **Test 4:** For each item that is initiated by the TOE but performed on the endpoint, the evaluator shall verify that the TOE requests the user's client to perform the action (generate a key and encrypt the data, zeroize a key).
- **Test 5:** For each method of changing a credential, the evaluator shall first provision the initial authorization factor(s) in the Enterprise Server, and then verify all authorization values supported allow the user access to the encrypted data on the user's client. Then the evaluator shall exercise the management functions to change the authorization factor values to a new one on the Enterprise Server. Then he or she will verify that the user's client denies access to the user’s encrypted data when he or she uses the old or original authorization factor values to gain access.
- **Test 6:** The evaluator shall add two administrators to the administrator group in the Enterprise Server and provision authorization factor(s) for each administrator. The evaluator shall verify that both administrators can log into the Enterprise Server using the provided authorization factors. The evaluator shall then exercise the management functions to change the authorization factor values for the first administrator to a new one on the Enterprise Server. Then he or she will verify that the Enterprise Server denies the first administrator access to the Management Console when the first administrator logs in with the old or original authorization factor to gain access. The evaluator shall also verify that the second administrator is still able to log in to the Enterprise Server with their original authorization factor.
- **Test 7:** The evaluator shall verify that the an administrator can configure each of the supported authorization factors attempts limits and shall verify that the user is denied access after surpassing that limit.
- **Test 8:** If the TOE provides the capability to disable management of any capability allowed in the EM PP-Module, the evaluator shall devise a test that ensures that each capability which can be disabled has been or can be disabled following guidance provided by the vendor.
- **Test 9:** If the TOE provides the capability to manage capabilities in place of the File Encryption Clients, where those administrative capabilities are then disabled in the File Encryption Clients, the evaluator shall devise a test that ensures that each capability which can be disabled in the File Encryption Clients and can be subsequently managed by the EM is tested as follows: Disable the administrative capability in

a File Encryption Client and enable it in the EM. Verify that the administration of the capability in the EM is successful.
- **Test 10:** The evaluator shall verify the encryption policy enforcement by changing the permitted algorithms and verifying the changes take place.

FMT_SMR.2 Restrictions on Security Roles

TSS
Refer to the evaluation activities for FMT_MOF.1.

Guidance
Refer to the evaluation activities for FMT_MOF.1.

Tests
Refer to the evaluation activities for FMT_MOF.1.

2.1.6 Protection of the TSF (FPT)

FPT_ITT.1 Basic Internal TSF Data Transfer Protection

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

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

Tests
The evaluator shall perform the following tests:

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

Further evaluation activities are associated with the specific protocols.

FPT_KYP_EXT.1 Protection of Keys and Key Material

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

KMD

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

Guidance
None.

Tests
None.

FPT_KYP_EXT.2 Attribution of Key and Key Material

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

Guidance

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

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

3 Evaluation Activities for Optional SFRs

The PP-Module does not define any optional requirements.

4 Evaluation Activities for Selection-Based SFRs

4.1 Cryptographic Support (FCS)

FCS_CKM_EXT.6 Cryptographic Password/Passphrase Conditioning

TSS
There are two aspects of this component that require evaluation: passwords/passphrases of the length specified in the requirement (at least 64 characters) are supported, and that the characters that are input are subject to the selected conditioning function. These activities are separately addressed in the text below.

Support for minimum length: The evaluator shall check to ensure that the TSS describes the allowable ranges for password/passphrase lengths, and that at least 64 characters may be specified by the user.

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

Support for PBKDF: The evaluator shall examine the TSS to ensure that the formation of all KEKs or FEKs (as decided in the FCS_CKM_EXT.3 selection) is described and that the key sizes match that described by the ST author.

The evaluator shall check that the TSS describes the method by which the password/passphrase is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length of the KEK selected in FCS_KYC_EXT.1.

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

FCS_CKM_EXT.6.2: The ST author shall provide a description in the TSS regarding the salt generation. The evaluator shall confirm that the salt is generated using an RBG described in FCS_RBG_EXT.1 (from the [AppPP]).

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

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

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

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

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

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

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

Tests
The evaluator shall perform the following tests:

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

4.2 Identification and Authentication (FIA)

FIA_CHR_EXT.1 Challenge/Response Recovery Credential

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

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

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

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

The evaluator shall perform the following tests:

- **Test 1:** The evaluator shall configure the Challenge/Response recovery to validate the user. The evaluator shall then issue a challenge on behalf of a managed user and ensure that TSF successfully generates the response.
- **Test 2:** The evaluator shall configure Challenge/Response recovery to validate the user. The evaluator shall then issue a challenge on behalf of managed User A and attempt to use it as an unmanaged User B to obtain a response. This should fail.
- **Test 3:** The evaluator shall issue a challenge on behalf of a managed user and ensure that the response received successfully will log the user in on that device.
- **Test 4:** The evaluator shall attempt to reuse the response of User A with User B on the same system and it should fail.
- **Test 5:** The evaluator shall issue a challenge on behalf of a managed user from a managed system, reboot the system [system terminates the session] and enter the response. This should fail.
- **Test 6:** The evaluator shall issue a challenge on behalf of a managed user and attempt to enter an incorrect response on the system the number of times described in the Guidance Documents. The observed behavior shall conform to the assignments/selections in FIA_CHR_EXT.1.5 and FIA_CHR_EXT.1.6.

4.3 Trusted Path/Channels (FTP)

FTP_TRP.1 Trusted Path

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

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

Tests
The evaluator shall perform the following tests:

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

Further evaluation activities are associated with the specific protocols.

5 Evaluation Activities for Objective SFRs

The PP-Module does not define any objective requirements.

6 Evaluation Activities for SARs

The PP-Module does not define any SARs beyond those defined within the App PP base to which it must claim conformance. It is important to note that a TOE that is evaluated against the PP-Module is inherently evaluated against this Base-PP as well. The App PP includes a number of Evaluation Activities associated with both SFRs and SARs. Additionally, the PP-Module includes a number of SFR-based Evaluation Activities that similarly refine the SARs of the Base-PPs. The evaluation laboratory will evaluate the TOE against the Base-PP and supplement that evaluation with the necessary SFRs that are taken from the PP-Module.

7 Required Supplementary Information

This Supporting Document has no required supplementary information beyond the ST, operational guidance, and testing.

Appendix A - References

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