Supporting Document Mandatory Technical Document

Evaluation Activities for collaborative Protection Profile for Hardcopy Devices

Version 0.8, 2020-11-18

Foreword

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

Supporting Documents 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 Supporting Document. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

This SD has been developed by the Hardcopy Devices International Technical Community (HCD-iTC) and is designed to be used to support the evaluations of TOEs against the cPP identified in Technology Area and Scope of Supporting Document.

Technical Editor

Hardcopy Device International Technical Community (HCD-iTC)

Revision History

Table 1. Revision history

Version	Date	Description
0.4	2020-08-26	Initial release for internal review
0.8	2020-11-18	Second release for HCD iTC review

General Purpose

See Technology Area and Scope of Supporting Document.

Field of special use

This Supporting Document applies to the evaluation of TOEs claiming conformance with the collaborative Protection Profile for Hardcopy Devices [HCDcPP].

Acknowledgements

This Supporting Document was developed by the HCD-iTC with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

Table of Contents

Chapter 1. Introduction

1.1. Technology Area and Scope of Supporting Document

This Supporting Document (SD) defines the Evaluation Activities (EAs) associated with the collaborative Protection Profile for Hardcopy Devices [HCDcPP].

This SD is mandatory for evaluations of products that claim conformance to any of the following cPP(s):

- collaborative Protection Profile for Hardcopy Devices, 0.6, 2020-06-08

Although EAs are defined mainly for the evaluator to follow, the definitions in this SD aim to provide a common understanding for developers, evaluators and users as to what aspects of the TOE are tested in an evaluation against the associated cPPs, and to what depth the testing is carried out. This common understanding in turn contributes to the goal of ensuring that evaluations against the cPP achieve comparable, transparent and repeatable results. In general, the definition of EAs will also help developers to prepare for evaluation by identifying specific requirements for their TOE. The specific requirements in EAs may in some cases clarify the meaning of SFRs, and may identify particular requirements for the content of Security Targets (STs) (especially the TOE Summary Specification (TSS)), Administrator Guidance Documentation (AGD), and possibly required supplementary information (e.g., for entropy analysis or cryptographic key management architecture - see Required Supplementary Information).

1.2. Structure of the Document

EAs can be defined for both SFRs and SARs. These are defined in separate sections of this SD.

If any EA 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 EA may be modified or deemed not applicable for a particular TOE, but this must be agreed with the Certification Body for the evaluation.

In general, if all EAs (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 EAs have been successfully completed would require a specific justification from the evaluator as to why the EAs were not sufficient for that TOE.

Similarly, at the more granular level of Assurance Components, if the EAs for an Assurance Component and all of its related SFR EAs 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 EAs have been successfully completed would require a specific justification from the evaluator as to why the EAs were not sufficient for that TOE.

Chapter 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 workunits that are performed in Section 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 workunit 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 workunits ADV_FSP.1-7, AGD_OPE.1-4, and AGD_OPE.1-5.

Finally, the subsection labelled Tests is where the iTC has 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 workunits 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. Security Audit (FAU)

2.1.1. FAU_GEN.1 Audit data generation

2.1.1.1. TSS

The evaluator shall check the TOE Summary Specification (TSS) to ensure that auditable events and its recorded information are consistent with the definition of the SFR.

2.1.1.2. Guidance Documentation

The evaluator shall check the guidance documents to ensure that auditable events and its recorded information are consistent with the definition of the SFRs.

2.1.1.3. Tests

The evaluator shall also perform the following tests:

The evaluator shall check to ensure that the audit record of each of the auditable events described in Table 3 of [HCDcPP] is appropriately generated.

The evaluator shall check a representative sample of methods for generating auditable events, if

there are multiple methods.

The evaluator shall check that FIA_UAU.1 events have been generated for each mechanism, if there are several different I&A mechanisms.

2.1.2. FAU_GEN.2 User identity association

The EAs for FAU_GEN.1 address this SFR.

2.1.3. FAU_STG_EXT.1 Extended: External Audit Trail Storage

2.1.3.1. TSS

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and “cleared” periodically by sending the data to the audit server.

2.1.3.2. Guidance Documentation

The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

2.1.3.3. Tests

The evaluator shall perform the following test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator’s choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

2.2. Cryptograhic Support (FCS)

2.2.1. FCS_CKM.1(b) Cryptographic key generation (Symmetric Keys)

2.2.1.1. TSS

The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked.

2.2.1.2. KMD

If the TOE is relying on random number generation from a third-party source, the KMD needs to describe the function call and parameters used when calling the third-party DRBG function. Also, the KMD needs to include a short description of the vendor's assumption for the amount of entropy seeding the third-party DRBG. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or the KMD to determine that the key size being requested is identical to the key size and mode to be used for the encryption/decryption of the user data (FCS_COP.1(d)).

The KMD is described in Appendix E of [HCDcPP].

2.2.2. FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

2.2.2.1. 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.

2.2.2.2. KMD

The evaluator shall verify the Key Management Description (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.4 for the destruction.

2.2.3. FCS_CKM.4 Cryptographic key destruction

2.2.3.1. TSS

The evaluator shall verify the TSS provides a high level description of how keys and key material are destroyed.

If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS 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 TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.

2.2.3.2. 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 nonvolatile memory) and how they are overwritten.

The evaluator shall check to ensure the KMD lists each type of key that is stored in nonvolatile memory, and identifies the memory type (volatile or nonvolatile) where key material is stored.

The KMD identifies and describes the interface(s) that is used to service commands to read/write memory. The evaluator examines the interface description for each different media type to ensure that the interface supports the selection(s) made by the ST Author.

2.2.3.3. Guidance Documentation

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 nonvolatile 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.

2.2.3.4. Tests

For these tests the evaluator shall utilize appropriate development environment (e.g., a Virtual

Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

Test 1: Applied to each key held as 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 nonvolatile memory). In the case where the only selection made for the destruction method key was removal of power, destruction of reference to the key directly followed by a request for garbage collection, or memory management, then this test is unnecessary. The evaluator shall:

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. 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: Applied to each key held in nonvolatile memory and subject to destruction by the TOE, except for replacing a key using the selection [a new value of a key of the same size]. 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 data 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. The test succeeds if step 3 fails.

Test 3: Applied to each key held in nonvolatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

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 nonvolatile memory 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 as nonvolatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

1. Record the 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. Search the storage location in Step #1 of nonvolatile 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.

2.2.4. FCS_COP.1(a) Cryptographic Operation (Symmetric encryption/decryption)

2.2.4.1. Tests

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The CMAC Validation System (CMACVS)", "The CCM Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS) with the Addition of XPN Validation Testing" (these documents are available from https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

2.2.5. FCS_COP.1(b) Cryptographic Operation (for signature generation/verification)

2.2.5.1. Tests

The evaluator shall use the signature generation and signature verification portions of "The FIPS 186-4 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The 186-4 RSA Validation System (RSA2VS)" as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-4). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

2.2.6. FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

2.2.6.1. TSS

For any RBG services provided by a third party, the evaluator shall ensure the TSS includes a statement about the expected amount of entropy received from such a source, and a full description of the processing of the output of the third-party source. The evaluator shall verify that this statement is consistent with the selection made in FCS_RBG_EXT.1.2 for the seeding of the DRBG. If the ST specifies more than one DRBG, the evaluator shall examine the TSS to verify that it identifies the usage of each DRBG mechanism.

2.2.6.2. Entropy Description

The evaluator shall ensure the Entropy Description provides all of the required information as described in Appendix D of [HCDcPP]. The evaluator assesses the information provided and ensures the TOE is providing sufficient entropy when it is generating a Random Bit String.

2.2.6.3. Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected DRBG mechanism(s), if necessary.

2.2.6.4. Tests

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable by the TOE, the evaluator shall perform 15 trials for each configuration. The evaluator shall verify that the instructions in the operational guidance for configuration of the RBG are valid.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “Generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the

personalization string lengths.

2.3. User Data Protection (FDP)

2.3.1. FDP_ACC.1 Subset access control

It is covered by assurance activities for FDP_ACF.1.

2.3.2. FDP_ACF.1 Security attribute based access control

2.3.2.1. TSS

The evaluator shall check to ensure that the TSS describes the functions to realize SFP defined in Table 4 and Table 5 of [HCDcPP].

2.3.2.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance contains a description of the operation to realize the SFP defined in Tables 4 and Table 5 of [HCDcPP], which is consistent with the description in the TSS.

2.3.2.3. Tests

The evaluator shall perform tests to confirm the functions to realize the SFP defined in Table 4 and Table 5 of [HCDcPP] with each type of interface (e.g., operation panel, Web interfaces) to the TOE.

The evaluator testing should include the following viewpoints:

- representative sets of the operations against representative sets of the object types defined in Table 4 and Table 5 of [HCDcPP] (including some cases where operations are either permitted or denied)
- representative sets for the combinations of the setting for security attributes that are used in access control

2.4. Identification and Authentication (FIA)

2.4.1. FIA_AFL.1 Authentication failure handling

2.4.1.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the actions in the case of authentication failure (types of authentication events, the number of unsuccessful authentication attempts, actions to be conducted), which is consistent with the definition of the SFR.

2.4.1.2. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance describes the setting for actions to be taken in the case of authentication failure, if any are defined in the SFR.

2.4.1.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the subsequent authentication attempts do not succeed by the behavior according to the actions defined in the SFR when unsuccessful authentication attempts reach the status defined in the SFR.
2. The evaluator shall check to ensure that authentication attempts succeed when conditions to re-enable authentication attempts are defined in the SFR and when the conditions are fulfilled.
3. The evaluator shall perform the tests 1 and 2 described above for all the targeted authentication methods when there are multiple Internal Authentication methods (e.g., password authentication, biometric authentication).
4. The evaluator shall perform the tests 1 and 2 described above for all interfaces when there are multiple interfaces (e.g., operation panel, Web interfaces) that implement authentication attempts.

2.4.2. FIA_ATD.1 User attribute definition

2.4.2.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the user security attributes that the TOE uses to implement the SFR, which is consistent with the definition of the SFR.

2.4.3. FIA_PMG_EXT.1 Extended: Password Management

2.4.3.1. Guidance Documentation

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of passwords, and that it provides instructions on setting the minimum password length.

2.4.3.2. Tests

The evaluator shall also perform the following test:

The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

2.4.4. FIA_UAU.1 Timing of authentication

2.4.4.1. TSS

The evaluator shall check to ensure that the TSS describes all the identification and authentication

mechanisms that the TOE provides (e.g., Internal Authentication and authentication by external servers).

The evaluator shall check to ensure that the TSS identifies all the interfaces to perform identification and authentication (e.g., identification and authentication from operation panel or via Web interfaces).

The evaluator shall check to ensure that the TSS describes the protocols (e.g., LDAP, Kerberos, OCSP) used in performing identification and authentication when the TOE exchanges identification and authentication with External Authentication servers.

The evaluator shall check to ensure that the TSS contains a description of the permitted actions before performing identification and authentication, which is consistent with the definition of the SFR.

2.4.4.2. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance contains descriptions of identification and authentication methods that the TOE provides (e.g., External Authentication, Internal Authentication) as well as interfaces (e.g., identification and authentication from operation panel or via Web interfaces), which are consistent with the ST (TSS).

2.4.4.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that identification and authentication succeeds, enabling the access to the TOE when using authorized data.
2. The evaluator shall check to ensure that identification and authentication fails, disabling the access to the TOE afterwards when using unauthorized data.

The evaluator shall perform the tests described above for each of the authentication methods that the TOE provides (e.g., External Authentication, Internal Authentication) as well as interfaces (e.g., identification and authentication from operation panel or via Web interfaces).

2.4.5. FIA_UAU.7 Protected authentication feedback

2.4.5.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the authentication information feedback provided to users while the authentication is in progress, which is consistent with the definition of the SFR.

2.4.5.2. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that only the information defined in the SFR is provided for feedback by attempting identification and authentication.
2. The evaluator shall perform the test 1 described above for all the interfaces that the TOE

provides (e.g., operation panel, identification and authentication via Web interface).

2.4.6. FIA_UID.1 Timing of identification

It is covered by assurance activities for FIA_UAU.1.

2.4.7. FIA_USB.1 User-subject binding

2.4.7.1. TSS

The evaluator shall check to ensure that the TSS contains a description of rules for associating security attributes with the users who succeed identification and authentication, which is consistent with the definition of the SFR.

2.4.7.2. Tests

The evaluator shall also perform the following test:

The evaluator shall check to ensure that security attributes defined in the SFR are associated with the users who succeed identification and authentication (it is ensured in the tests of FDP_ACF) for each role that the TOE supports (e.g., User and Administrator).

2.5. Security Management (FMT)

2.5.1. FMT_MOF.1 Management of security functions behavior

2.5.1.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the management functions that the TOE provides as well as user roles that are permitted to manage the functions, which is consistent with the definition of the SFR.

The evaluator shall check to ensure that the TSS identifies interfaces to operate the management functions.

2.5.1.2. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance describes the operation methods for users of the given roles defined in the SFR to operate the management functions.

2.5.1.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can operate the management functions in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected.
3. The evaluator shall check to ensure that U.NORMAL is not permitted to operate the

management functions.

2.5.2. FMT_MSA.1 Management of security attributes

2.5.2.1. TSS

The evaluator shall check to ensure that the TSS contains a description of possible operations for security attributes and given roles to those security attributes, which is consistent with the definition of the SFR.

2.5.2.2. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance contains a description of possible operations for security attributes and given roles to those security attributes, which is consistent with the definition of the SFR.

The evaluator shall check to ensure that the administrator guidance describes the timing of modified security attributes.

2.5.2.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can perform operations to the security attributes in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected as specified in the administrator guidance.
3. The evaluator shall check to ensure that a user that is not part of an authorized role defined in the SFR is not permitted to perform operations on the security attributes.

2.5.3. FMT_MSA.3 Static attribute initialization

2.5.3.1. TSS

The evaluator shall check to ensure that the TSS describes mechanisms to generate security attributes which have properties of default values, which are defined in the SFR.

2.5.3.2. Tests

If U.ADMIN is selected, then testing of this SFR is performed in the tests of FDP_ACF.1.

2.5.4. FMT_MTD.1 Management of TSF data

2.5.4.1. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance identifies the management operations and authorized roles consistent with the SFR.

The evaluator shall check to ensure that the administrator guidance describes how the assignment of roles is managed.

The evaluator shall check to ensure that the administrator guidance describes how security attributes are assigned and managed.

The evaluator shall check to ensure that the administrator guidance describes how the security-related rules (.e.g., access control rules, timeout, number of consecutive logon failures,) are configured.

2.5.4.2. Tests

The evaluator shall perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can perform operations to TSF data in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected as specified in the administrator guidance.
3. The evaluator shall check to ensure that no users other than users of the given roles defined in the SFR can perform operations to TSF data.

2.5.5. FMT_SMF.1 Specification of Management Functions

2.5.5.1. TSS

The evaluator shall check the TSS to ensure that the management functions are consistent with the assignment in the SFR.

2.5.5.2. Guidance Documentation

The evaluator shall check the guidance documents to ensure that management functions are consistent with the assignment in the SFR, and that their operation is described.

2.5.6. FMT_SMR.1 Security roles

2.5.6.1. TSS

The evaluator shall check to ensure that the TSS contains a description of security related roles that the TOE maintains, which is consistent with the definition of the SFR.

2.5.6.2. Tests

As for tests of this SFR, it is performed in the tests of FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1.

2.6. Protection of the TSF (FPT)

2.6.1. FPT_SKP_EXT.1 Extended: Protection of TSF Data

2.6.1.1. TSS

The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

2.6.2. FPT_STM.1 Reliable time stamps

2.6.2.1. TSS

The evaluator shall check to ensure that the TSS describes mechanisms that provide reliable time stamps.

2.6.2.2. Guidance Documentation

The evaluator shall check to ensure that the guidance describes the method of setting the time.

2.6.2.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the time is correctly set up in accordance with the guidance or external network services (e.g., NTP).
2. The evaluator shall check to ensure that the time stamps are appropriately provided.

2.6.3. FPT_TST_EXT.1 Extended: TSF testing

2.6.3.1. TSS

The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

2.6.3.2. Guidance Documentation

The evaluator shall also ensure that the operational guidance describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

2.6.4. FPT_TUD_EXT.1 Extended: Trusted Update

2.6.4.1. TSS

The evaluator shall check to ensure that the TSS contains a description of mechanisms that verify software for update when performing updates, which is consistent with the definition of the SFR.

The evaluator shall check to ensure that the TSS identifies interfaces for administrators to obtain the current version of the TOE as well as interfaces to perform updates.

2.6.4.2. Guidance Documentation

The evaluator shall check to ensure that the administrator guidance contains descriptions of the operation methods to obtain the TOE version as well as the operation methods to start update processing, which are consistent with the description of the TSS.

2.6.4.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure the current version of the TOE can be appropriately obtained by means of the operation methods specified by the administrator guidance.
2. The evaluator shall check to ensure that the verification of the data for updates of the TOE succeeds using authorized data for updates by means of the operation methods specified by the administrator guidance.
3. The evaluator shall check to ensure that only administrators can implement the application for updates using authorized data for updates.
4. The evaluator shall check to ensure that the updates are correctly performed by obtaining the current version of the TOE after the normal updates finish.
5. The evaluator shall check to ensure that the verification of the data for updates of the TOE fails using unauthorized data for updates by means of the operation methods specified by the administrator guidance. (The evaluator shall also check those cases where hash verification mechanism, if selected in FPT_TUD_EXT.1.3, and digital signature verification mechanism fail.)

2.7. TOE Access (FTA)

2.7.1. FTA_SSL.3 TSF-initiated termination

2.7.1.1. TSS

The evaluator shall check to ensure that the TSS describes the types of user sessions to be terminated (e.g., user sessions via operation panel or Web interfaces) after a specified period of user inactivity.

2.7.1.2. Guidance Documentation

The evaluator shall check to ensure that the guidance describes the default time interval and, if it is settable, the method of setting the time intervals until the termination of the session.

2.7.1.3. Tests

The evaluator shall also perform the following tests:

1. If it is settable, the evaluator shall check to ensure that the time until the termination of the session can be set up by the method of setting specified in the administrator guidance.
2. The evaluator shall check to ensure that the session terminates after the specified time interval.
3. The evaluator shall perform the tests 1 and 2 described above for all the user sessions identified in the TSS.

2.8. Trusted Channels (FTP)

2.8.1. FTP_ITC.1 Inter-TSF trusted channel

2.8.1.1. TSS

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

2.8.1.2. Tests

The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.
3. The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.
4. The evaluator shall ensure, 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 assurance activities are associated with the specific protocols.

2.8.2. FTP_TRP.1(a) Trusted path (for Administrators)

2.8.2.1. 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.

2.8.2.2. Guidance Documentation

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

2.8.2.3. Tests

The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
3. The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.

Further assurance activities are associated with the specific protocols.

Chapter 3. Evaluation Activities for Conditionally Mandatory Requirements

3.1. Confidential Data on Field-Replaceable Nonvolatile Storage Devices

3.1.1. FPT_KYP_EXT.1 Extended: Protection of Key and Key Material

3.1.1.1. KMD

The evaluator shall examine the Key Management Description (KMD) for a description of the methods used to protect keys stored in nonvolatile memory.

The evaluator shall verify the KMD to ensure it describes the storage location of all keys and the protection of all keys stored in nonvolatile memory.

3.1.2. FCS_KYC_EXT.1 Extended: Key Chaining

3.1.2.1. TSS

The evaluator shall verify the TSS contains a high-level description of the BEV sizes – that it supports BEV outputs of no fewer 128 bits for products that support only AES-128, and no fewer than 256 bits for products that support AES-256.

3.1.2.2. KMD

The evaluator shall examine the KMD to ensure that it describes a high level description of the key hierarchy for all accepted BEVs. The evaluator shall examine the KMD to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap, submask combining, or key encryption.

The evaluator shall examine the KMD 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. (e.g., using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the effective strength of the BEV is maintained throughout the key chain.

The evaluator shall verify the KMD includes a description of the strength of keys throughout the key chain.

3.1.3. FDP_DSK_EXT.1 Extended: Protection of Data on Disk

In the EAs, below, “Device” refers to the Field-Replaceable Nonvolatile Storage Device from FDP_DSK_EXT.1. If the TOE contains more than one applicable Device, then the EAs are performed

as necessary on each such Device.

3.1.3.1. TSS

If the self-encrypting device option is selected, the Device must be certified in conformance to the current Full Disk Encryption Protection Profile. The tester shall confirm that the specific SED is listed in the TSS, documented and verified to be CC certified against the FDE EE cPP.

The evaluator shall examine the TSS to ensure that the description is comprehensive in how the data is written to the Device and the point at which the encryption function is applied.

For the cryptographic functions that are provided by the Operational Environment, the evaluator shall check the TSS to ensure it describes the interface(s) used by the TOE to invoke this functionality.

The evaluator shall verify that the TSS describes the initialization of the Device at shipment of the TOE, or by the activities the TOE performs to ensure that it encrypts all the storage devices entirely when a user or administrator first provisions the Device. The evaluator shall verify the TSS describes areas of the Device that it does not encrypt (e.g., portions that do not contain confidential data boot loaders, partition tables, etc.). If the TOE supports multiple Device encryptions, the evaluator shall examine the administration guidance to ensure the initialization procedure encrypts all Devices.

3.1.3.2. Guidance Documentation

The evaluator shall review the AGD guidance to determine that it describes the initial steps needed to enable the Device encryption function, including any necessary preparatory steps. The guidance shall provide instructions that are sufficient to ensure that all Devices will be encrypted when encryption is enabled or at shipment of the TOE.

3.1.3.3. KMD

The evaluator shall verify the KMD includes a description of the data encryption engine, its components, and details about its implementation (e.g., for hardware: integrated within the device's main SOC or separate co-processor, for software: initialization of the Device, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and areas which are not encrypted (e.g., boot loaders, portions that do not contain confidential data, partition tables, etc.)). The evaluator shall verify the KMD provides a functional (block) diagram showing the main components (such as memories and processors) and the data path between, for hardware, the Device's interface and the Device's persistent media storing the data, or for software, the initial steps needed to the activities the TOE performs to ensure it encrypts the storage device entirely when a user or administrator first provisions the product. The hardware encryption diagram shall show the location of the data encryption engine within the data path. The evaluator shall validate that the hardware encryption diagram contains enough detail showing the main components within the data path and that it clearly identifies the data encryption engine.

The evaluator shall verify the KMD provides sufficient instructions to ensure that when the encryption is enabled, the TOE encrypts all Field-Replaceable Nonvolatile Storage Devices. The evaluator shall verify that the KMD describes the data flow from the interface to the Device's persistent media storing the data. The evaluator shall verify that the KMD provides information on

those conditions in which the data bypasses the data encryption engine (e.g., read-write operations to an unencrypted area).

The evaluator shall verify that the KMD provides a description of the boot initialization, the encryption initialization process, and at what moment the product enables the encryption. If encryption can be enabled and disabled, the evaluator shall validate that the product does not allow for the transfer of confidential data before it fully initializes the encryption. The evaluator shall ensure the software developer provides special tools which allow inspection of the encrypted drive either in-band or out-of-band, and may allow provisioning with a known key.

3.1.3.4. Tests

The evaluator shall perform the following tests:

Test 1. Write data to Storage device: Perform writing to the storage device with operating TSFI which enforce write process of User documents and Confidential TSF data.

Test 2. Confirm that written data are encrypted: Verify there are no plaintext data present in the encrypted range written by Test 1; and, verify that the data can be decrypted by proper key and key material.

All TSFIs for writing User Document Data and Confidential TSF data should be tested by above Test 1 and Test 2.

3.2. PSTN Fax-Network Separation

3.2.1. FDP_FXS_EXT.1 Extended: Fax separation

The following assurance activities are required when the TOE has a fax communication function to transmit and receive via PSTN.

3.2.1.1. TSS

The evaluator shall check the TSS to ensure that it describes:

1. The fax interface use cases
2. The capabilities of the fax modem and the supported fax protocols
3. The data that is allowed to be sent or received via the fax interface
4. How the TOE can only be used transmitting or receiving User Data using fax protocols

3.2.1.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance contains a description of the fax interface in terms of usage and available features.

3.2.1.3. Tests

The evaluator shall test to ensure that the fax interface can only be used transmitting or receiving User Data using fax protocols. Testing will be dependent upon how the TOE enforces this

requirement. The following tests shall be used and supplemented with additional testing or a rationale as to why the following tests are sufficient:

1. Verify that the TOE accepts incoming calls using fax carrier protocols and rejects calls that use data carriers. For example, this may be achieved using a terminal application to issue modem commands directly to the TOE from a PC modem (issue terminal command: 'ATDT <TOE Fax Number>') – the TOE should answer the call and disconnect.
2. Verify TOE negotiates outgoing calls using fax carrier protocols and rejects negotiation of data carriers. For example, this may be achieved by using a PC modem to attempt to receive a call from the TOE (submit a fax job from the TOE to <PC modem number>, at PC issue terminal command: 'ATA') – the TOE should disconnect without negotiating a carrier.

3.3. Network Communications

3.3.1. FTP_TRP.1(b) Trusted path (for Non-administrators)

3.3.1.1. TSS

The evaluator shall examine the TSS to determine that the methods of remote TOE access for non-administrative users are indicated, along with how those communications are protected.

The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE access are consistent with those specified in the requirement, and are included in the requirements in the ST.

3.3.1.2. Guidance Documentation

The evaluator shall confirm that the operational guidance contains instructions for establishing the remote user sessions for each supported method.

3.3.1.3. Tests

The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each specified (in the operational guidance) remote user access method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each method of remote access supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote user session without invoking the trusted path.
3. The evaluator shall ensure, for each method of remote user access, the channel data are not sent in plaintext.

Further EAs are associated with the specific protocols.

Chapter 4. Evaluation Activities for Optional Requirements

4.1. Internal Audit Log Storage

4.1.1. FAU_SAR.1 Audit review

4.1.1.1. TSS

The evaluator shall check to ensure that the TSS contains a description that audit records can be viewed only by an Administrator and functions to view audit records.

The evaluator shall check to ensure that the TSS contains a description of the methods of using interfaces that retrieve audit records (e.g., methods for user identification and authentication, authorization, and retrieving audit records).

4.1.1.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance appropriately describes the ways of viewing audit records and forms of viewing.

4.1.1.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the forms of audit records are provided as specified in the operational guidance by retrieving audit records in accordance with the operational guidance.
2. The evaluator shall check to ensure that no users other than an Administrator can retrieve audit records.
3. The evaluator shall check to ensure that all audit records are retrieved by the operation of retrieving audit records.

4.1.2. FAU_SAR.2 Restricted audit review

4.1.2.1. Tests

The evaluator shall include test 2 related to this function in the set of tests performed in FAU_SAR.1.

4.1.3. FAU_STG.1 Protected audit trail storage

4.1.3.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the means of preventing audit records from unauthorized access (modification, deletion).

4.1.3.2. Guidance Documentation

The evaluator shall check to ensure that the TSS and operational guidance contain descriptions of the interfaces to access to audit records, and if the descriptions of the means of preventing audit records from unauthorized access (modification, deletion) are consistent.

4.1.3.3. Tests

The evaluator shall also perform the following test:

1. The evaluator shall test that an authorized user can access the audit records.
2. The evaluator shall test that a user without authorization for the audit data cannot access the audit records.

4.1.4. FAU_STG.4 Prevention of audit data loss

4.1.4.1. TSS

The evaluator shall check to ensure that the TSS contains a description of the processing performed when the capacity of audit records becomes full, which is consistent with the definition of the SFR.

4.1.4.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance contains a description of the processing performed (such as informing the authorized users) when the capacity of audit records becomes full.

4.1.4.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator generates auditable events after the capacity of audit records becomes full by generating auditable events in accordance with the operational guidance.
2. The evaluator shall check to ensure that audit records are processed in accordance with the definition of the SFR.

4.2. Image Overwrite

4.2.1. FDP_RIP.1(a) Subset residual information protection

4.2.1.1. TSS

The evaluator shall examine the TSS to ensure that the description is comprehensive in describing where image data is stored and how and when it is overwritten.

4.2.1.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance contains instructions for enabling the Image Overwrite function.

4.2.1.3. Tests

The evaluator shall include tests related to this function in the set of tests performed in FMT_SMF.1.

4.3. Purge Data

4.3.1. FDP_RIP.1(b) Subset residual information protection

4.3.1.1. TSS

The evaluator shall examine the TSS to ensure that the description is comprehensive in describing what customer-supplied data is to be purged, where it is stored, and how it is made unavailable.

4.3.1.2. Guidance Documentation

The evaluator shall check to ensure that the operational guidance contains instructions for initiating the Purge Data function.

4.3.1.3. Tests

The evaluator shall include tests related to this function in the set of tests performed in FMT_SMF.1.

4.4. Asymmetric Key Generation

4.4.1. FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys)

4.4.1.1. TSS

The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described in the TSS.

The TSS may refer to the KMD, described in Appendix E of [HCDcPP], that may not be made available to the public.

4.4.1.2. Tests

The evaluator shall use the key pair generation portions of "The FIPS 186-4 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and “The 186-4 RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

Chapter 5. Evaluation Activities for Selection-Based Requirements

5.1. Confidential Data on Field-Replaceable Nonvolatile Storage Devices

5.1.1. FCS_COP.1(d) Cryptographic operation (AES Data Encryption/Decryption)

5.1.1.1. TSS

The evaluator shall verify the TSS includes a description of the key size used for encryption and the mode used for encryption.

5.1.1.2. Guidance Documentation

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

5.1.1.3. Tests

The following tests are conditional based upon the selections made in the SFR.

AES-CBC Tests

AES-CBC Known Answer Tests

There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for

encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.

KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key
    for i = 1 to 1000:
            if i == 1:
                CT[1] = AES-CBC-Encrypt(Key, IV, PT)
                PT = IV
            else:
                CT[i] = AES-CBC-Encrypt(Key, PT)
                PT = CT[i-1]
```

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Test

The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

128 bit and 256 bit keys

Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

XTS-AES Test The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

256 bit (for AES-128) and 512 bit (for AES-256) keys

Three data unit (i.e., plaintext) lengths. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 2^{16} bits, whichever is smaller.

The evaluator shall test the encrypt functionality using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

5.1.2. FCS_COP.1(e) Cryptographic operation (Key Wrapping)

5.1.2.1. TSS

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.

5.1.2.2. 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.

5.1.2.3. Tests

The evaluator shall ensure the wrapped key is wrapped as specified in this SFR using reference implementation of wrapping in accordance with AES in the modes and key size specified in this SFR. This reference implementation of wrapping algorithm may be a tool or program provided by the evaluator or the developer, this implementation is dependent on the KMD description provided by the developer.

5.1.3. FCS_COP.1(f) Cryptographic operation (Key Encryption)

5.1.3.1. TSS

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

5.1.3.2. KMD

The evaluator shall review the KMD to ensure that all keys are encrypted using the approved method and a description of when the key encryption occurs is provided.

5.1.3.3. Tests

The evaluator shall use tests in FCS_COP.1(d) to verify encryption.

5.1.4. FCS_COP.1(i) Cryptographic operation (Key Transport)

5.1.4.1. TSS

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.

5.1.4.2. Guidance Documentation

There are no AGD evaluation activities for this SFR.

5.1.4.3. KMD

There are no KMD evaluation activities for this SFR.

5.1.4.4. 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:

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.
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.

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 decryptions where the A value is passed incorrectly, and the evaluator shall verify that an error is detected.
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.
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:

1. The evaluator shall inspect each cipher text, C, produced by RSA-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 RSA-KEMKWS decryption operation a cipher text of unsupported length and verify that an error is detected and that the decryption process stops.
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 RSA-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 RSA-KEM-KWS decryption process detects the error.
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 RSA-KEM-KWS decryption process given in NIST 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 RSA-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 RSA-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 RSA-

KEM-KWS decryption operation and verify that a decryption error is detected.

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.

5.1.5. FCS_SMC_EXT.1 Extended: Submask Combining

5.1.5.1. TSS

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 DEK.

5.1.5.2. KMD

The evaluator shall review the KMD to ensure that an approved combination is used and does not result in the weakening or exposure of key material.

5.1.5.3. Tests

(conditional): If there is more than one authorization factor, the evaluator shall ensure that failure to supply a required authorization factor does not result in access to the encrypted data.

5.2. Protected Communications

5.2.1. FCS_IPSEC_EXT.1 Extended: IPsec selected

5.2.1.1. TSS

5.2.1.1.1. FCS_IPSEC_EXT.1.1

The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet) and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular

packet) as well as packets that are part of an established SA.

5.2.1.1.2. FCS_IPSEC_EXT.1.2

The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as selected).

5.2.1.1.3. FCS_IPSEC_EXT.1.3

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

5.2.1.1.4. FCS_IPSEC_EXT.1.4

The evaluator shall examine the TSS to verify that the symmetric encryption algorithms selected (along with the SHA-based HMAC algorithm, if AES-CBC is selected) are described. If selected, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(g) Cryptographic Operations (for keyed-hash message authentication).

5.2.1.1.5. FCS_IPSEC_EXT.1.5

The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

5.2.1.1.6. FCS_IPSEC_EXT.1.6

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

5.2.1.1.7. FCS_IPSEC_EXT.1.7

The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

5.2.1.1.8. FCS_IPSEC_EXT.1.9

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

5.2.1.1.9. FCS_IPSEC_EXT.1.10

The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the signature algorithm or algorithms specified in the requirement.

5.2.1.2. Guidance Documentation

5.2.1.2.1. FCS_IPSEC_EXT.1.1

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the guidance documentation is consistent with the description in the TSS, and that the level of detail in the guidance documentation is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

5.2.1.2.2. FCS_IPSEC_EXT.1.2

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

5.2.1.2.3. FCS_IPSEC_EXT.1.3

The evaluator checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.

5.2.1.2.4. FCS_IPSEC_EXT.1.4

The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the algorithms selected by the ST author.

5.2.1.2.5. FCS_IPSEC_EXT.1.5

The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test if IKEv2 is selected.

5.2.1.2.6. FCS_IPSEC_EXT.1.6

The evaluator ensures that the operational guidance describes the configuration of the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test for each ciphersuite selected.

5.2.1.2.7. FCS_IPSEC_EXT.1.7

If the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

5.2.1.2.8. FCS_IPSEC_EXT.1.8

The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the values allow for Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

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

5.2.1.3. Tests

5.2.1.3.1. FCS_IPSEC_EXT.1.1

The evaluator uses the guidance documentation to configure the TOE to carry out the following tests:

a) Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and (if configurable) allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g., a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.

b) Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.

5.2.1.3.2. FCS_IPSEC_EXT.1.2

The evaluator shall perform the following test(s) based on the selections chosen:

1. (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures an IPsec Peer to operate in tunnel mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the client to connect to the IPsec Peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.
2. (conditional): If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures an IPsec Peer to operate in transport mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowed

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

5.2.1.3.3. FCS_IPSEC_EXT.1.3

The evaluator shall perform the following test:

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

5.2.1.3.4. FCS_IPSEC_EXT.1.4

The evaluator shall also perform the following tests:

The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the selected algorithms, and attempt to establish a connection using ESP. The connection should be successfully established for each algorithm.

5.2.1.3.5. FCS_IPSEC_EXT.1.5

(conditional): If IKEv2 is selected, the evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

5.2.1.3.6. FCS_IPSEC_EXT.1.6

The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

5.2.1.3.7. FCS_IPSEC_EXT.1.7

The evaluator shall also perform the following test:

(conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.

5.2.1.3.8. FCS_IPSEC_EXT.1.8

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

1. (Conditional): The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is renegotiated.
2. (Conditional): The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.
3. (Conditional): The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

5.2.1.3.9. FCS_IPSEC_EXT.1.9

The evaluator shall also perform the following test (this test may be combined with other tests for this component, for instance, the tests associated with FCS_IPSEC_EXT.1.1):

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

5.2.1.3.10. FCS_IPSEC_EXT.1.10

The evaluator shall also perform the following test:

For each supported signature algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved and results in the successful establishment of a connection.

5.2.2. FCS_TLS_EXT.1 Extended: TLS selected

5.2.2.1. TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

5.2.2.2. Tests

The evaluator shall also perform the following test:

1. The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a

ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

2. The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic:
    a. [Conditional: TOE is a server] Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.
    b. [Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
    c. [Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange.
    d. [Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.

5.2.3. FCS_SSH_EXT.1 Extended: SSH selected

5.2.3.1. TSS

5.2.3.1.1. FCS_SSH_EXT.1.2

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed.

5.2.3.1.2. FCS_SSH_EXT.1.4

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

5.2.3.1.3. FCS_SSH_EXT.1.5

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

5.2.3.1.4. FCS_SSH_EXT.1.6

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE

(specifically, that the “none” MAC algorithm is not allowed).

5.2.3.2. Guidance Documentation

5.2.3.2.1. FCS_SSH_EXT.1.7

The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is “hard-coded” into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol.

5.2.3.3. Tests

5.2.3.3.1. FCS_SSH_EXT.1.2

The evaluator shall also perform the following tests:

1. The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.
2. Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

5.2.3.3.2. FCS_SSH_EXT.1.3

The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

5.2.3.3.3. FCS_SSH_EXT.1.4

The evaluator shall also perform the following test:

The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

5.2.3.3.4. FCS_SSH_EXT.1.5

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

5.2.3.3.5. FCS_SSH_EXT.1.6

The evaluator shall also perform the following test:

The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

5.2.3.3.6. FCS_SSH_EXT.1.7

The evaluator shall also perform the following test:

1. [Conditional: TOE is a client] The evaluator shall configure an SSH server to permit all allowed key exchange methods. For each allowed key exchange method, the evaluator shall attempt to connect from the TOE to the SSH server and observe that the attempt succeeds.
2. [Conditional: TOE is a server] The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.
3. [Conditional: TOE is a server] For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.

5.2.4. FCS_HTTPS_EXT.1 Extended: HTTPS selected

5.2.4.1. TSS

5.2.4.1.1. FCS_HTTPS_EXT.1.2

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack.

5.2.4.2. Tests

5.2.4.2.1. FCS_HTTPS_EXT.1.2

Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

5.2.5. FCS_COP.1(g) Cryptographic Operation (for keyed-hash message authentication)

5.2.5.1. Tests

The evaluator shall use "The Keyed-Hash Message Authentication Code Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

5.2.6. FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

5.2.6.1. TSS

The evaluator shall examine the TSS to ensure that it states that text-based pre-shared keys of 22 characters are supported, and that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to

the bit string used by IPsec, and that this conditioning is consistent with the first selection in the FIA_PSK_EXT.1.3 requirement. If the assignment is used to specify conditioning, the evaluator will confirm that the TSS describes this conditioning.

If “bit-based pre-shared keys” is selected, the evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

5.2.6.2. Guidance Documentation

The evaluator shall examine the operational guidance to determine that it provides guidance on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

5.2.6.3. Tests

The evaluator shall also perform the following tests:

1. The evaluator shall compose at least 15 pre-shared keys of 22 characters that cover all allowed characters in various combinations that conform to the operational guidance, and demonstrates that a successful protocol negotiation can be performed with each key.
2. [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.
3. [conditional]: If the TOE supports bit-based pre-shared keys but does not generate such keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
4. [conditional]: If the TOE supports bit-based pre-shared keys and does generate such keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

5.3. Trusted Update

5.3.1. FCS_COP.1(c) Cryptographic operation (Hash Algorithm)

5.3.1.1. TSS

The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

5.3.1.2. Guidance Documentation

The evaluator checks the operational guidance documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present.

5.3.1.3. Tests

The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented test mode.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

Short Messages Test Bit-oriented Mode

The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test Byte-oriented Mode

The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test Bit-oriented Mode

The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 99*i, where $1 \leq i \leq m$. For SHA-512, the length of the i-th message is 1024 + 99*i, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test Byte-oriented Mode

The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 8*99*i, where $1 \leq i \leq m/8$. For SHA-512, the length of the i-th message is 1024 + 8*99*i, where $1 \leq i \leq m/8$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Pseudorandomly Generated Messages Test

This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of "The Secure Hash Algorithm Validation System (SHAVS)". The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

5.4. Passphrase-based Key Entry

5.4.1. FCS_PCC_EXT.1 Extended: Cryptographic Password Construct and Conditioning

5.4.1.1. TSS

The evaluator shall ensure the TSS describes the manner in which the TOE enforces the construction of passwords, including the length, and requirements on characters (number and type). The TSS also provides a description of how the password is conditioned and the evaluator ensures it satisfies the requirement.

5.4.1.2. KMD

The evaluator shall examine the KMD to ensure that the formation of the BEV and intermediary keys is described and that the key sizes match that selected by the ST Author.

The evaluator shall check that the KMD 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 KMD 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 as the BEV as specified above.

5.4.1.3. Tests

The evaluator shall also perform the following tests:

Test 1: Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.

Test 2: If the TOE supports a password/passphrase length up to a maximum number of characters, n (which would be greater than 64), then ensure that the TOE will not accept more than n characters.

Test 3: Ensure that the TOE supports passwords consisting of all characters assigned and supported by the ST author.

5.4.2. FCS_KDF_EXT.1 Extended: Cryptographic Key Derivation

5.4.2.1. TSS

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.

5.4.2.2. 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.

5.4.3. FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication)

5.4.3.1. TSS

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

5.4.3.2. Tests

For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be equal to the result of generating HMAC tags with the same key using a known good implementation.

5.4.4. FCS_SNI_EXT.1 Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

5.4.4.1. TSS

The evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generating using an RBG described in FCS_RBG_EXT.1.

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.

Chapter 6. Evaluation Activities for SARs

The sections below specify EAs for the Security Assurance Requirements (SARs) included in the related cPPs. The EAs in Evaluation Activities for SFRs, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements, and Evaluation Activities for Selection-Based Requirements are an interpretation of the more general CEM assurance requirements as they apply to the specific technology area of the TOE.

In this section, each SAR that is contained in the cPP is listed, and the EAs that are not associated with an SFR are captured here, or a reference is made to the CEM, and the evaluator is expected to perform the CEM work units.

6.1. Class ASE: Security Target

When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator ensures the content of the TSS in the ST satisfies the EAs specified in Evaluation Activities for SFRs as well as the EAs for the conditionally-mandatory, optional, and selection-based SFRs claimed by the ST and specified in Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements, and Evaluation Activities for Selection-Based Requirements.

6.2. Class ADV: Development

6.2.1. Basic Functional Specification (ADV_FSP.1)

The EAs for this assurance component focus on understanding the interfaces (e.g., application programing interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Evaluation Activities for SFRs, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements, Evaluation Activities for Selection-Based Requirements, and in EAs for AGD, ATE and AVA SARs in other parts of Evaluation Activities for SARs.

The EAs are reworded (indicated with italicization) for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional "functional specification" documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no

separate mapping information is required for this element.

Table 2. Mapping of ADV_FSP.1 CEM Work Units to Evaluation Activities

CEM ADV_FSP.1 Work Units	Evaluator Activities
ADV_FSP.1-1 The evaluator ***shall examine*** the functional specification to determine that it states the purpose of each SFR-supporting and SFR-enforcing TSFI.	ADV_FSP.1-1 Evaluation Activity: *The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.*
ADV_FSP.1-2 The evaluator ***shall examine*** the functional specification to determine that the method of use for each SFR-supporting and SFR-enforcing TSFI is given.	ADV_FSP.1-2 Evaluation Activity: *The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.*
ADV_FSP.1-3 The evaluator ***shall examine*** the presentation of the TSFI to determine that it identifies all parameters associated with each SFR-enforcing and SFR supporting TSFI.	ADV_FSP.1-3 Evaluation Activity: *The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.*
ADV_FSP.1-4 The evaluator ***shall examine*** the rationale provided by the developer for the implicit categorisation of interfaces as SFR-non-interfering to determine that it is accurate.	Paragraph 561 from the CEM: "In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, this work unit should be considered satisfied." Since the rest of the ADV_FSP.1 work units will have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.
ADV_FSP.1-5 The evaluator ***shall check*** that the tracing links the SFRs to the corresponding TSFIs.	ADV_FSP.1-5 Evaluation Activity: *The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.*
ADV_FSP.1-6 The evaluator ***shall examine*** the functional specification to determine that it is a complete instantiation of the SFRs.	EAs that are associated with the SFRs in Evaluation Activities for SFRs, and, if applicable, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements and Evaluation Activities for Selection-Based Requirements, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are covered. Therefore, the intent of this work unit is covered.

CEM ADV_FSP.1 Work Units	Evaluator Activities
ADV_FSP.1-7 The evaluator ***shall examine*** the functional specification to determine that it is an accurate instantiation of the SFRs.	EAs that are associated with the SFRs in Evaluation Activities for SFRs, and, if applicable, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements and Evaluation Activities for Selection-Based Requirements, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are addressed, and that the description of the interfaces is accurate with respect to the specification captured in the SFRs. Therefore, the intent of this work unit is covered.

6.2.2. ADV_FSP.1-1 Evaluation Activity

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g., audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent, is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

6.2.3. ADV_FSP.1-2 Evaluation Activity

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g., audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent, is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

6.2.4. ADV_FSP.1-3 Evaluation Activity

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

6.2.5. ADV_FSP.1-5 Evaluation Activity

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Sections 2 and, if applicable, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements and Evaluation Activities for Selection-Based Requirements, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have an interface that is explicitly "mapped" to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

6.3. Class AGD: Guidance Documentation

It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the EAs in this section are described under the traditionally separate AGD families, the mapping between the documentation provided by the developer and the AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to administrators and users (as appropriate) as part of the TOE.

6.3.1. Operational User Guidance (AGD_OPE.1)

The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR.

In addition, the evaluator performs the EAs specified below.

6.3.1.1. Evaluation Activity

The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

6.3.1.2. Evaluation Activity

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately

address all platforms claimed for the TOE in the Security Target.

6.3.1.3. Evaluation Activity

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

6.3.1.4. Evaluation Activity

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

6.3.1.5. Evaluation Activity

In addition the evaluator shall ensure that the following requirements are also met.

1. The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.
2. The documentation must describe the process for verifying updates to the TOE by verifying a digital signature. The evaluator shall verify that this process includes the following steps:
    a. Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
    b. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.
3. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the EAs.

6.3.2. Preparative Procedures (AGD_PRE.1)

The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

Preparative procedures are distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

In addition, the evaluator performs the EAs specified below.

6.3.2.1. Evaluation Activity

The evaluator shall examine the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

6.3.2.2. Evaluation Activity

The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

6.3.2.3. Evaluation Activity

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

6.3.2.4. Evaluation Activity

The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

6.3.2.5. Evaluation Activity

In addition the evaluator shall ensure that the following requirements are also met.

The preparative procedures must

1. Include instructions to provide a protected administrative capability; and
2. Identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

6.4. Class ALC: Life-cycle Support

6.4.1. Labelling of the TOE (ALC_CMC.1)

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

6.4.2. TOE CM coverage (ALC_CMS.1)

When evaluating the developer’s coverage of the TOE in their CM system, the evaluator performs

the work units as presented in the CEM.

6.5. Class ATE: Tests

6.5.1. Independent Testing - Conformance (ATE_IND.1)

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Evaluation Activities for SFRs, Evaluation Activities for Conditionally Mandatory Requirements, Evaluation Activities for Optional Requirements, and Evaluation Activities for Selection-Based Requirements.

The evaluator should consult Equivalency Considerations when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

6.6. Class AVA: Vulnerability Assessment

6.6.1. Vulnerability Survey (AVA_VAN.1)

While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator follows a set of well-defined activities, and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis, and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

In order to meet these goals some refinement of the AVA_VAN.1 CEM work units is needed. The following table indicates, for each work unit in AVA_VAN.1, whether the CEM work unit is to be performed as written, or if it has been clarified by an Evaluation Activity. If clarification has been provided, a reference to this clarification is provided in the table.

Table 3. Mapping of AVA_VAN.1 CEM Work Units to Evaluation Activities

CEM AVA_VAN.1 Work Units	Evaluator Activities
AVA_VAN.1-1 The evaluator ***shall examine*** the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.	The evaluator shall perform the CEM activity as specified. *If the iTC specifies any tools to be used in performing this analysis in section A.3.4, the following text is also included in this cell: "The calibration of test resources specified in paragraph 1418 of the CEM applies to the tools listed in Appendix A, Section A.1.4."*

CEM AVA_VAN.1 Work Units	Evaluator Activities
AVA_VAN.1-2 The evaluator ***shall examine*** the TOE to determine that it has been installed properly and is in a known state. The evaluator shall perform the CEM activity as specified.	The evaluator shall perform the CEM activity as specified.
AVA_VAN.1-3 The evaluator ***shall examine*** sources of information publicly available to identify potential vulnerabilities in the TOE. Replace CEM work unit with activities outlined in Appendix A, Section A.1	Replace CEM work unit with activities outlined in Appendix A, Section A.1
AVA_VAN.1-4 The evaluator ***shall record*** in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.	Replace the CEM work unit with the analysis activities on the list of potential vulnerabilities in Appendix A, section A.1, and documentation as specified in Appendix A, Section A.3.
AVA_VAN.1-5 The evaluator ***shall devise*** penetration tests, based on the independent search for potential vulnerabilities.	Replace the CEM work unit with the activities specified in Appendix A, section A.2.

CEM AVA_VAN.1 Work Units	Evaluator Activities
AVA_VAN.1-6 The evaluator ***shall produce*** penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include: a. Identification of the potential vulnerability the TOE is being tested for; b. Instructions to connect and setup all required test equipment as required to conduct the penetration test; c. Instructions to establish all penetration test prerequisite initial conditions; d. Instructions to stimulate the TSF; e. Instructions for observing the behaviour of the TSF; f. Descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results; g. Instructions to conclude the test and establish the necessary post-test state for the TOE.	The CEM work unit is captured in Appendix A, Section A.3; there are no substantive differences.
AVA_VAN.1-7 The evaluator shall conduct penetration testing.	The evaluator shall perform the CEM activity as specified. See Appendix A, Section A.3, paragraph 110 for guidance related to attack potential for confirmed flaws.
AVA_VAN.1-8 The evaluator shall record the actual results of the penetration tests.	The evaluator shall perform the CEM activity as specified.

CEM AVA_VAN.1 Work Units	Evaluator Activities
AVA_VAN.1-9 The evaluator shall report in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.	Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.
AVA_VAN.1-10 The evaluator shall examine the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Basic attack potential.	This work unit is not applicable for Type 1 and Type 2 flaws (as defined in Appendix A, Section A.1), as inclusion in this Supporting Document by the iTC makes any confirmed vulnerabilities stemming from these flaws subject to an attacker possessing a Basic attack potential. This work unit is replaced for Type 3 and Type 4 flaws by the activities defined in Appendix A, Section A.2, paragraph 110.
AVA_VAN.1-11 The evaluator shall report in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each: a. Its source (e.g., CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication); b. The SFR(s) not met; c. A description; d. Whether it is exploitable in its operational environment or not (i.e., exploitable or residual). e. The amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.	Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.

Because of the level of detail required for the evaluation activities, the bulk of the instructions are

contained in Appendix A, while an "outline" of the assurance activity is provided below.

6.6.1.1. Evaluation Activity (Documentation)

GUIDANCE:

If the iTC determines that no additional documentation beyond that specified below is required, it is acceptable to remove this Evaluation Activity in the Supporting Document.

If the iTC determines that additional documentation is appropriate, they will insert a description of that documentation in this paragraph. The iTC must specify the required documentation in as much detail as possible to eliminate issues associated with the evaluators evaluating the suitability of the documentation rather than using the documentation to evaluate the product. Therefore, documentation statements such as "Supply a high-level and low-level design" are discouraged. An example of a better statement is:

"The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components apply to all systems claimed in the ST, and should identify at a minimum the processors used by the TOE. Software components include any libraries used by the TOE, such as cryptographic libraries. This additional documentation is merely a list of the name and version number of the components, and will be used by the evaluators in formulating hypotheses during their analysis."

The evaluator shall examine the documentation outlined below provided by the vendor to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

In addition to the activities specified by the CEM in accordance with Table 2 above, the evaluator shall perform the following activities.

6.6.1.2. Evaluation Activity

GUIDANCE:

The evaluator formulates hypotheses in accordance with process defined in Appendix A.1. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Chapter 7. Required Supplementary Information

This Supporting Document refers in various places to the possibility that 'required supplementary information' may need to be supplied as part of the deliverables for an evaluation. This term is intended to describe information that is not necessarily included in the Security Target or operational guidance, and that may not necessarily be public. Examples of such information could be entropy analysis, or description of a cryptographic key management architecture used in (or in support of) the TOE. The requirement for any such supplementary information will be identified in the relevant cPP.

The supplementary information required by this SD are:

- Entropy Documentation, which is evaluated against the guidance specified in Appendix D of [HCDcPP].
- A Key Management Description (KMD), which is evaluated against guidance specified in Appendix E of [HCDcPP] and all relevant KMD Evaluation Activities in this SD.

Chapter 8. References

- [2600] IEEE Std. 2600™-2008 “IEEE Standard for Information Technology: Hardcopy Device and System Security”
- [2600.1] IEEE Std. 2600.1™-2009 “IEEE Standard for a Protection Profile in Operational Environment A”
- [610.12] IEEE Std 610.12-1990 “IEEE Standard Glossary of Software Engineering Terminology”
- [8802-6] ISO /IEC 8802-6:1994 “Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 6”
- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1 Revision 5, April 2017.
- [addenda] CC and CEM addenda, Exact Conformance, Selection-Based SFRs, Optional SFRs, Version 0.5, May 2017
- [HCDcPP] collaborative Protection Profile for Hardcopy Devices, 0.6, 2020-06-08

Appendix A: Vulnerability Analysis

A.1. Sources of Vulnerability Information

CEM Work Unit AVA_VAN.1-3 has been supplemented in this Supporting Document to provide a better-defined set of flaws to investigate and procedures to follow based on this particular technology. Terminology used is based on the flaw hypothesis methodology, where the evaluation team hypothesizes flaws and then either proves or disproves those flaws (a flaw is equivalent to a "potential vulnerability" as used in the CEM). Flaws are categorized into four "types" depending on how they are formulated:

1. A list of flaw hypotheses applicable to the technology described by the cPP derived from public sources as documented in Type 1 Hypotheses - Public-Vulnerability-based-this fixed set has been agreed to by the iTC. Additionally, this will be supplemented with entries for a set of public sources (as indicated below) that are directly applicable to the TOE or its identified components (as defined by the process in Type 1 Hypotheses - Public-Vulnerability-based below); this is to ensure that the evaluators include in their assessment applicable entries that have been discovered since the cPP was published;
2. A list of flaw hypotheses contained in this document that are derived from lessons learned specific to that technology and other iTC input (that might be derived from other open sources and vulnerability databases, for example) as documented in Type 2 Hypotheses - iTC-sourced;
3. A list of flaw hypotheses derived from information available to the evaluators; this includes the baseline evidence provided by the vendor described in this Supporting Document (documentation associated with EAs, documentation described in Evaluation Activity (Documentation), <the iTC can remove the reference to Evaluation Activity (Documentation) if no additional documentation is defined> documentation described in Required Supplementary Information), as well as other information (public and/or based on evaluator experience) as documented in Type 3 Hypotheses - Evaluation-Team-Generated; and
4. A list of flaw hypotheses that are generated through the use of iTC-defined tools (e.g., nmap, protocol testers) and their application is specified in Type 4 Hypotheses - Tool-Generated.

GUIDANCE:

It should be noted that it is not mandatory for the Supporting Document to contain all types of flaw hypotheses listed above; that determination is left to the iTC. Should the iTC determine there are no applicable flaws for a given type, they should adjust the text of the Supporting Document accordingly to remove spurious sections and references.

A.1.1. Type 1 Hypotheses - Public-Vulnerability-based

The iTC must determine what public vulnerability databases are to be used as the basis for Type 1 hypotheses, and what entries in these databases apply. A sample list of resources is contained in Appendix D, but the iTC is not bound by that list.

In performing this activity, the iTC first agrees upon the sources to be used. The list of sources should be searched by the iTC with an agreed-upon set of terms such that the iTC feels a representative set of vulnerabilities with respect to the technology type is returned.

Having identified the sources, for each source the iTC defines criteria for selecting entries in the list. The lists and criteria should be identified in this section of the Supporting Document so that evaluators can use the same sources and criteria at evaluation time to select entries that were made after the cPP was published. For each entry that meets the criteria, the iTC determines whether or not to include it in the list of flaw hypotheses defined in this Supporting Document. This will likely necessitate the creation of some criteria by which to judge an entry that is agreed to by the iTC. For instance, CVEs that would generate flaw hypotheses related to buffer overflows would probably be rejected as a generic flaw hypothesis. The output of this activity is a list of specific entries from the selected sources that will be used as flaw hypotheses.

The following list of public sources of vulnerability information was selected by the iTC:

REVIEW:

list of sources; can be an appendix (for example, see Appendix D) or it can be an in-line bulleted list. The references should be specific enough so there is no ambiguity identifying a specific location/database.

The list of sources above was searched with the following search terms:

REVIEW:

list of search terms used by the iTC; if search terms/criteria are specific to specific sources, the association between the terms used and the source should be made here as well.

In order to supplement this list, the evaluators shall also perform a search on the sources listed above to determine a list of potential flaw hypotheses that are more recent that the publication date of the cPP, and those that are specific to the TOE and its components as specified by the additional documentation mentioned above. Any duplicates - either in a specific entry, or in the flaw hypothesis that is generated from an entry from the same or a different source - can be noted and removed from consideration by the evaluation team.

As part of type 1 flaw hypothesis generation for the specific components of the TOE, the evaluator shall also search the component manufacturer’s websites to determine if flaw hypotheses can be generated on this basis (for instance, if security patches have been released for the version of the component being evaluated, the subject of those patches may form the basis for a flaw hypothesis).

GUIDANCE:

The iTC should insert any additional instructions to the evaluators for performing searches of public sources of vulnerability information applicable to the technology that is not covered above.

A.1.2. Type 2 Hypotheses - iTC-sourced

The iTC must consider if there are any technology-specific vulnerabilities or types of vulnerabilities that the evaluators should consider that are not contained in the previous section. This could be based on previous evaluations against the cPP, experience of the iTC members, or other factors. These vulnerabilities should be limited to those exploitable with a Basic Attack Potential-as characterized by the time, technical expertise, knowledge of the TOE, equipment, and access needed for exploitation. Section B.4.2.2. of the CEM provides detailed guidance on how these factors should be considered in determining attack potential relative to vulnerabilities.

This set of vulnerabilities (Type 2) is listed below and would then need to be considered by the evaluation team. It is likely that there will be few or no entries identified for this type until more experience is gained with the cPP.

The following list of flaw hypothesis generated by the iTC for this technology must be considered by the evaluation team as flaw hypotheses in performing the vulnerability assessment.

REVIEW:

List of flaw hypotheses generated by the iTC as indicated above.

If the evaluators discover a Type 3 or Type 4 flaw that they believe should be considered as a Type 2 flaw in future versions of this cPP, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

A.1.3. Type 3 Hypotheses - Evaluation-Team-Generated

Type 3 flaws are formulated by the evaluator based on information presented by the product (through on-line help, product documentation and user guides, etc.) and product behaviour during the (functional) testing activities. The evaluator is also free to formulate flaws that are based on material that is not part of the baseline evidence (e.g., information gleaned from an Internet mailing list, or reading interface documentation on interfaces not included in the set provided by the developer), although such activities have the potential to vary significantly based upon the product and evaluation facility performing the analysis.

If the evaluators discover a Type 3 flaw that they believe should be considered as a Type 2 flaw in future versions of this cPP, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

GUIDANCE:

It may be the case that no activities of this type are appropriate for this technology; in that case, this section can be removed and references in other areas of this Supporting Document adjusted accordingly.

A.1.4. Type 4 Hypotheses - Tool-Generated

The iTC identifies any tools to be used or testing to be performed by the evaluators resulting in the creation of flaw hypotheses. The iTC can choose to merely outline the testing that needs to be formed, or they can identify specific tools and testing to be done with those tools. In this definition, the iTC also indicates test results that indicate that a flaw hypothesis should be created (the goal of this section is not to perform or re-do functional testing; it is to test in a way that might produce anomalies that are to be candidate flaw hypotheses). The iTC documents and specifies tools; the procedures, settings, and testing to be performed; and criteria for creating flaw hypotheses from these results in this section.

It may be the case that no activities of these type are appropriate for this technology; in that case, this section can be removed and references in other areas of this Supporting Document adjusted accordingly.

If the evaluators discover a Type 4 flaw that they believe should be considered as a Type 2 flaw in future versions of this cPP, they should work with their Certification Body to determine the appropriate means of submitting the flaw for consideration by the iTC.

A.2. Process for Evaluator Vulnerability Analysis

As flaw hypotheses are generated from the activities described above, the evaluation team will disposition them; that is, attempt to prove, disprove, or determine the non-applicability of the hypotheses. This process is as follows.

The evaluator will refine each flaw hypothesis for the TOE and attempt to disprove it using the information provided by the developer or through penetration testing. During this process, the evaluator is free to interact directly with the developer to determine if the flaw exists, including requests to the developer for additional evidence (e.g., detailed design information, consultation with engineering staff); however, the CB should be included in these discussions. Should the developer object to the information being requested as being not compatible with the overall level of the evaluation activity/cPP and cannot provide evidence otherwise that the flaw is disproved, the evaluator prepares an appropriate set of materials as follows:

1. The source documents used in formulating the hypothesis, and why it represents a potential compromise against a specific TOE function;
2. An argument why the flaw hypothesis could not be proven or disproved by the evidence provided so far; and
3. The type of information required to investigate the flaw hypothesis further.

The Certification Body (CB) will then either approve or disapprove the request for additional information. If approved, the developer provides the requested evidence to disprove the flaw hypothesis (or, of course, acknowledge the flaw).

For each hypothesis, the evaluator will note whether the flaw hypothesis has been successfully disproved, successfully proven to have identified a flaw, or requires further investigation. It is important to have the results documented as outlined in Section A.3 below.

If the evaluator finds a flaw, the evaluator must report these flaws to the developer. All reported flaws must be addressed as follows:

If the developer confirms that the flaw exists and that it is exploitable at Basic Attack Potential, then a change is made by the developer, and the resulting resolution is agreed by the evaluator and noted as part of the evaluation report.

If the developer, the evaluator, and the CB agree that the flaw is exploitable only above Basic Attack Potential and does not require resolution for any other reason, then no change is made and the flaw is noted as a residual vulnerability in the CB-internal report (ETR).

If the developer and evaluator agree that the flaw is exploitable only above Basic Attack Potential, but it is deemed critical to fix because of technology-specific or cPP-specific aspects such as typical use cases or operational environments, then a change is made by the developer, and the resulting resolution is agreed by the evaluator and noted as part of the evaluation report.

Disagreements between evaluator and vendor regarding questions of the existence of a flaw, its attack potential, or whether it should be deemed critical to fix are resolved by the CB.

Any testing performed by the evaluator shall be documented in the test report as outlined in Reporting below.

As indicated in Reporting, Reporting, the public statement with respect to vulnerability analysis that is performed on TOEs conformant to the cPP is constrained to coverage of flaws associated with Types 1 and 2 (defined in Sources of Vulnerability Information) flaw hypotheses only. The fact that the iTC generates these candidate hypotheses indicates these must be addressed.

For flaws of Types 3 and 4, each CB is responsible for determining what constitutes Basic Attack Potential for the purposes of determining whether a flaw is exploitable in the TOE's environment. The determination criteria shall be documented in the CB-internal report as specified in Reporting. As this is a per-CB activity, no public claims are made with respect to the resistance of a particular TOE against flaws of Types 3 and 4; rather, the claim is that the activities outlined in this appendix were carried out, and the evaluation team and CB agreed that any residual vulnerabilities are not exploitable by an attacker with Basic Attack Potential.

A.3. Reporting

The evaluators shall produce two reports on the testing effort; one that is public-facing (that is, included in the non-proprietary evaluation report, which is a subset of the Evaluation Technical Report (ETR)), and the complete ETR that is delivered to the overseeing CB.

The public-facing report contains:

- The flaw identifiers returned when the procedures for searching public sources were followed according to instructions in the Supporting Document per Type 1 Hypotheses - Public-Vulnerability-based;
- A statement that the evaluators have examined the Type 1 flaw hypotheses specified in this Supporting Document in Type 1 Hypotheses - Public-Vulnerability-based (i.e., the flaws listed in the previous bullet) and the Type 2 flaw hypotheses specified in this Supporting Document by the iTC in Type 2 Hypotheses - iTC-sourced.

GUIDANCE:

The above two bullets encompass all flaw hypotheses of Type 1 and Type 2.

- A statement that the evaluation team developed Types 3 and 4 flaw hypotheses in accordance with Sections Type 3 Hypotheses - Evaluation-Team-Generated, Type 4 Hypotheses - Tool-Generated, and Process for Evaluator Vulnerability Analysis, and that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential as defined by the CB in accordance with the guidance in the CEM. It should be noted that this is just a statement about the "fact of" Types 3 and 4 flaw hypotheses being developed, and that no specifics about the number of flaws, the flaws themselves, or the analysis pertaining to those flaws will be included in the public-facing report.

No other information is provided in the public-facing report.

The internal CB report contains, in addition to the information in the public-facing report:

- A list of all of the flaw hypotheses generated (cf. AVA_VAN.1-4);
- The evaluator penetration testing effort, outlining the testing approach, configuration, depth and results (cf. AVA_VAN.1-9);
- All documentation used to generate the flaw hypotheses (in identifying the documentation used in coming up with the flaw hypotheses, the evaluation team must characterize the documentation so that a reader can determine whether it is strictly required by this Supporting Document, and the nature of the documentation (design information, developer engineering notebooks, etc.));
- The evaluator shall report all exploitable vulnerabilities and residual vulnerabilities, detailing for each:
  - Its source (e.g., CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);
  - The SFR(s) not met;
  - A description;
  - Whether it is exploitable in its operational environment or not (i.e., exploitable or residual).
  - The amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities (cf. AVA_VAN.1-11);
- How each flaw hypothesis was resolved (this includes whether the original flaw hypothesis was confirmed or disproved, and any analysis relating to whether a residual vulnerability is

exploitable by an attacker with Basic Attack Potential) (cf. AVA_VAN1-10); and

- In the case that actual testing was performed in the investigation (either as part of flaw hypothesis generation using tools specified by the iTC in Section A.1.4, or in proving/disproving a particular flaw) the steps followed in setting up the TOE (and any required test equipment); executing the test; post-test procedures; and the actual results (to a level of detail that allow repetition of the test, including the following:
  - Identification of the potential vulnerability the TOE is being tested for;
  - Instructions to connect and setup all required test equipment as required to conduct the penetration test;
  - Instructions to establish all penetration test prerequisite initial conditions;
  - Instructions to stimulate the TSF;
  - Instructions for observing the behaviour of the TSF;
  - Descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;
  - Instructions to conclude the test and establish the necessary post-test state for the TOE. (cf. AVA_VAN.1-6, AVA_VAN.1-8).

Appendix B: Equivalency Considerations

GUIDANCE:

If the cPP does not require definitions of equivalency, then this section can be removed. Otherwise this section should be updated based on the iTC considerations for how a vendor can define equivalent products and how a lab should treat the definitions.

B.1. Introduction

This appendix provides a foundation for evaluators to determine whether a vendor's request for equivalency of products is allowed.

For the purpose of this evaluation, equivalency can be broken into two categories:

- **Variations in models**: Separate TOE models/variations may include differences that could necessitate separate testing across each model. If there are no variations in any of the categories listed below, the models may be considered equivalent.
- **Variations in TOE dependencies on the environment (e.g., OS/platform the product is tested on)**: The method a TOE provides functionality (or the functionality itself) may vary depending upon the environment on which it is installed. If there is no difference in the TOE-provided functionality or in the manner in which the TOE provides the functionality, the models may be considered equivalent.

Determination of equivalency for each of the above specified categories can result in several different testing outcomes.

If a set of TOE are determined to be equivalent, testing may be performed on a single variation of the TOE. However, if the TOE variations have security-relevant functional differences, each of the TOE models that exhibits either functional or structural differences must be separately tested. Generally speaking, only the difference between each variation of TOE must be separately tested. Other equivalent functionality may be tested on a representative model and not across multiple platforms.

If it is determined that a TOE operates the same regardless of the environment, testing may be performed on a single instance for all equivalent configurations. However, if the TOE is determined to provide environment-specific functionality, testing must take place in each environment for which a difference in functionality exists. Similar to the above scenario, only the functionality affected by environment differences must be retested.

If a vendor disagrees with the evaluator's assessment of equivalency, the Scheme arbitrates between the two parties whether equivalency exists.

B.2. Evaluator guidance for determining equivalence

B.2.1. Strategy

When performing the equivalency analysis, the evaluator should consider each factor independently. A factor may be any number of things at various levels of abstraction, ranging from the processor a device uses, to the underlying operating system and hardware platform a software application relies upon. Examples may be the various chip sets employed by the product, the type of network interface (different device drivers), storage media (solid state drive, spinning disk, EEPROM). It is important to consider how the difference in these factors may influence the TOE's ability to enforce the SFRs. Each analysis of an individual factor will result in one of two outcomes:

- For the particular factor, all variations of the TOE on all supported platforms are equivalent. In this case, testing may be performed on a single model in a single test environment and cover all supported models and environments.
- For the particular factor, a subset of the product has been identified to require separate testing to ensure that it operates identically to all other equivalent TOEs. The analysis would identify the specific combinations of models/testing environments that needed to be tested.

Complete CC testing of the product would encompass the totality of each individual analysis performed for each of the identified factors.

B.2.2. Test presentation/Truth in advertising

In addition to determining what to test, the evaluation results and resulting validation report must identify the actual module and testing environment combinations that have been tested. The analysis used to determine the testing subset may be considered proprietary and will only optionally be publicly included.

Appendix C: Public Vulnerability Sources

REVIEW:

This appendix may be included (if referenced from Appendix A above) or excluded from the final Supporting Document at the discretion of the iTC. The iTC should also consider supplementing it (if retained) as necessary to reflect the appropriate technology-specific aspects.

The following sources of public vulnerabilities are sources for the iTC to consider in both formulating the specific list of flaws to be investigated by the evaluators, as well as to reference in directing the evaluators to perform key-word searches during the evaluation of a specific TOE.

1. Search Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
2. Search Core Security Technologies: http://www.coresecurity.com
3. Search eEye Digital Security: http://blog.beyondtrust.com/zd_threat?status=zeroday
4. Search Exploit / Vulnerability Search Engine: www.exploitsearch.net
5. Conduct SecurITeam Exploit Search: www.securiteam.com
6. Search SecurityTracker: www.securitytracker.com
7. Search VUPEN Security, formerly FrSIRT: www.vupen.com
8. Conduct Google search: www.google.com
9. Search McAfee Threat Intelligence http://www.mcafee.com/us/mcafee-labs/threat-intelligence.aspx
10. Search Open Source Vulnerability Database http://osvdb.org/
11. Search Secwatch Advisories & Exploits https://securitynewsportal.com/index.shtml
12. Search Symantec http://www.symantec.com/security_response/
13. Search Tenable Network Security http://nessus.org/plugins/index.php?view=search
14. Tipping Point Zero Day Initiative http://www.zerodayinitiative.com/advisories
15. Search US-CERT http://www.kb.cert.org/vuls/html/search
16. Search Vigil@nce http://vigilance.fr/

Appendix D: Glossary

For definitions of standard CC terminology see [CC1].

Administrator

A User who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the security policies of the TOE. Administrators may possess special privileges that provide capabilities to override portions of security policies. [2600.1]

Confidential (TSF) Data

Assets for which either disclosure or alteration by a User who is not an Administrator or the owner of the data would have an effect on the operational security of the TOE. [2600.1]

DEK

A key used to encrypt data at rest.

External Authentication

Identification and authentication mechanism that uses services of External IT Entities to authenticate TOE Users.

Field-Replaceable (Unit)

The smallest subassembly that can be swapped in the field to repair a fault. [IEEE]

Hardcopy Device

A system producing or utilizing a physical embodiment of an electronic document or image. These systems include printers, scanners, fax machines, digital copiers, MFPs (multifunction peripherals), MFDs (multifunction devices), “all-in-ones” and other similar products.[2600]

Intermediate Key

A key used in a point between the initial user authorization and the DEK. [CPP_FDE_EE_V2.0]

Internal Authentication

Identification and authentication function that is wholly contained within the TOE.

Job

A document processing task submitted to the hardcopy device. A single processing task may process one or more documents. [2600.1]

Nonvolatile Storage Device

A device that provides computer storage of data that is not cleared when the power is turned off.

Operational Environment

Environment in which the TOE is operated.[CC]

Protection Profile

Implementation-independent statement of security needs for a TOE type. [CC]

Read

To access data from a storage device or data medium. (Note that in this case, the data medium may be a printed output, and therefore, release of a print job is a “read” operation) [610.12]

Security Assurance Requirement

A description of how assurance is to be gained that the TOE meets the SFRs. [CC]

Security Functional Requirement

A translation of the Security Objectives for the TOE into a standardized language. [CC]

Security Objective

Statement of an intent to counter identified Threats and/or satisfy identified organization security policies and/or Assumptions. [CC]

Security Target

Implementation-dependent statement of security needs for a specific identified TOE. [CC]

Submask

A submask is a bit string that can be generated and stored in a numbers of ways, such as passphrases, tokens, etc. [CPP_FDE_EE_V2.0]

Target of Evaluation

Set of software, firmware and/or hardware possibly accompanied by guidance. [CC]

TOE Security Functionality

Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs. [CC]

TSF Data

Data for the operation of the TOE upon which the enforcement of the SFR relies. [CC]

Unauthorized Access

Access to a resource that a User is not permitted to access.

User

Human or IT entity possibly interacting with the TOE from outside of the TOE boundary. [CC]

User Data

Data for the User that does not affect the operation of the TSF. [CC]

User Document Data

The Asset that consists of the information contained in a User’s Document. This includes the original Document itself in either hardcopy or electronic form, image data, or residually stored data created by the hardcopy device while processing an original Document and printed hardcopy output. [2600.1]

Appendix E: Acronyms

Table 4. Acronyms

Acronym	Meaning
AAD	Additional Authenticated Data
AES-CBC	Advanced Encryption Standard (AES) Cipher Algorithm in Cipher Block Chaining (CBC) Mode
AES-CCM	Advanced Encryption Standard (AES) Cipher Algorithm in Counter with CBC-MAC (CCM) Mode
AES-GCM	Advanced Encryption Standard (AES) Cipher Algorithm in Galois/Counter Mode (GCM)
XTS-AES	Advanced Encryption Standard (AES) Cipher Algorithm in XEX-based tweaked-codebook mode with ciphertext stealing (XTS)
AES	Advanced Encryption Standard
AESAVS	The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)
ATA	Advanced Technology Attachment
ASCII	American Standard Code for Information Interchange
BEV	Border Encryption Value
CC	Common Criteria
CCM	Counter with CBC-MAC
CCMVS	The CCM Validation System
CEM	Common Evaluation Methodology
CMAC	Cipher-based Message Authentication Code
CMACVS	The CMAC Validation System (CMACVS)
CSP	Critical Security Parameter
CVE	Common Vulnerabilities and Exposures
cPP	collaborative Protection Profile
DEK	Data Encryption Key
DHE	Diffie-Hellman Ephemeral
DRBG	Deterministic Random Bit Generator
DSA2VS	The FIPS 186-4 Digital Signature Algorithm Validation System (DSA2VS)
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA2VS	The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)
EEPROM	Electrically Erasable Programmable Read Only Memory
ETR	Evaluation Technical Report

Acronym	Meaning
ESP	Encapsulating Security Payload
FDE	Full-Disk Encryption
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
GCMVS	The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS) with the Addition of XPN Validation Testing
GMAC	Galois Message Authentication Code
HCD	Hardcopy Device
HMAC	Keyed-Hashing for Message Authentication
HMACVS	The Keyed-Hash Message Authentication Code Validation System (HMACVS)
IPsec	Internet Protocol Security
IT	Information Technology
IV	Initialization Vector
I&A	Identification and Authentication
HTTPS	Hypertext Transfer Protocol Secure
IKE	Internet Key Exchange
KAT	Known Answer Test
KMD	Key Management Description
KWK	KEK Wrapping Key
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OAEP	Optimal Asymmetric Encryption Padding
OCSP	Online Certificate Status Protocol
PC	Personal Computer
PP	Protection Profile
PSTN	Public Switched Telephone Network
RAID	Redundant Array of Independent Disks
RBG	Random Bit Generator
RFC	Request for Comments
RSA	Rivest–Shamir–Adleman
RSA2VS	The 186-4 RSA Validation System (RSA2VS)

Acronym	Meaning
SAR	Security Assurance Requirement
SED	Self-Encrypting Drive
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHAVS	The Secure Hash Algorithm Validation System (SHAVS)
SOC	System on a Chip
SPD	Security Problem Definition
SSH	Secure Shell
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Functionality
TSFI	TOE Security Functional Interface
TSS	TOE Summary Specification
UDP	User Datagram Protocol
VPN	Virtual Private Network