Assurance Activity Report

VID 11041

20-3516-R-0001 V1.1

February 7, 2020

ASURRE-Stor™ Solid State Self-Encrypting Drive

Security Target for Mercury Systems ASURRE-Stor™ Solid State Self-Encrypting Drive

Version 1.1, February 6, 2020

Evaluated by:

UL Verification Services Inc.
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

TOE Evaluation Sponsor and Developer:
Mercury Systems, Inc.
3601 E University Dr
Phoenix, AZ 85034

ST Author:
UL Verification Services Inc.
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401

Evaluation Personnel:
Oleg Andrianov
Gerrit Kruitbosch
Lucas Shaffer

Applicable Common Criteria Version
CC Version 3.1 R5, April 2017

Common Evaluation Methodology Version
CEM Version 3.1 R5, April 2017

Applicable Common Criteria Protection Profiles
collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201, February 1, 2019

collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0 + Errata 20190201, February 1, 2019

Table of Contents

1 Overview

This document presents evaluation results of the Mercury Systems ASURRE-Stor™ Solid State Self-Encrypting Drives the collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201, February 1, 2019 and the collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0 + Errata 20190201, February 1, 2019. This document contains a description of the assurance activities and associated results as performed by UL Verification Services Inc., an accredited Common Criteria Testing Laboratory. This Evaluation was conducted with the oversight and guidance provided by the National Information Assurance Partnership and its contributors.

1.1 Test Equivalency

The evaluator performed testing on the model ASD256AM2R running firmware revision 1.5.1 and hardware revision 3. The [ST] claims 4 models that differ only in addressable storage capacity, overwise they are equivalent. All testing was performed on the ASD256AM2R, except for FPT_DSK_EXT.1, that was performed an all models.

1.2 Test Environment

The test environment used by the CCTL during the course of testing is briefly summarized below and conforms to the expected use-case of the TOE (encrypted storage).

The evaluation team performed the independent testing activities to confirm the TOE operates to the TOE security functional requirements as specified in the [ST] for a product claiming conformance to [PP1] and [PP2]. The evaluation team devised a Test Plan based on the Testing Assurance Activities specified in [PP1] and [PP2]. The Test Plan described how each test activity was to be performed. The evaluation team executed the tests specified in the Test Plan and documented the results in the Evaluation Technical Report. The evaluation team consisted of Gerrit Kruitbosch from the CCTL.

The test laboratory was configured by UL and physically located at Mercury Systems, Inc's. facility in an access controlled room. The CCTL verified that all test evidence supported the conclusions of the test report.

1.2.1 Test Equipment:

ASRock Motherboard based Custom Desktop Computer

- Windows 7 Enterprise
- MDU (Mercury Drive Utility) Version 1.5.1
- AHCDEMO -- AHCI Demo Program -- Version 1B2

1.2.2 Test Scripts

- changeFirmware.tcl
- readSectors.tcl
- ataUserPassword.tcl
- WRANDOM.SC

1.2.3 Test Firmware:

- Firmware.error.bin
- Firmware.monsoon256A.1.5.1.bin

1.2.4 Initial Configuration

The evaluator configured the TOE according to the initial evaluated configuration described in the guidance documentation in Section 18.

2 SFR Assurance Activities and Results

2.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

TSS

The evaluator shall first examine the TSS to ensure that the authorization factors specified in the ST are described. For password-based factors the examination of the TSS section is performed as part of FCS_PCC_EXT.1 Evaluation Activities. Additionally in this case, the evaluator shall verify that the operational guidance discusses the characteristics of external authorization factors (e.g., how the authorization factor must be generated; format(s) or standards that the authorization factor must meet) that are able to be used by the TOE.

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

Guidance

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

KMD

The evaluator shall examine the Key Management Description to confirm that the initial authorization factors (submasks) directly contribute to the unwrapping of the BEV.

The evaluator shall verify the KMD describes how a submask is produced from the authorization factor (including any associated standards to which this process might conform), and verification is performed to ensure the length of the submask meets the required size (as specified in this requirement).

Test

The password authorization factor is tested in FCS_PCC_EXT.1.

The evaluator shall also perform the following tests:

Test 1 (conditional): If there is more than one authorization factor, ensure that failure to supply a required authorization factor does not result in access to the decrypted plaintext data.

2.1.1 TSS

[ST] Section 7.1 states that only one authorization factor is supported, a user-supplied password composed of up to 64 characters. [AGD] Section 15 states that the TOE accepts passwords of up to 64 characters.

2.1.2 Guidance

[AGD] Section 28 states there are only 2 modes that the TOE can operate in that are CC-Compliant. These are Mode 1 and Mode 6, both of which only use a password as an authorization factor. This section also states a Crypto Officer shall enforce a password of at least 8 characters.

2.1.3 KMD

[KMD] Section 6, Figures 2 and 3 show that a user password is conditioned using a PBKDF HMAC-SHA512 to unwrap the BEV (Mode 1) or to create an intermediate key that will unwrap the BEV.

[ST] Section 7.1.1 describes how the submask is being produced from the authorization factor and ensures its length by using a password buffer.

2.1.4 Testing

N/A: the TOE only supports a password authorization factor.

2.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

TSS

The evaluator shall examine the TSS for a description of authorization factors and which of the factors are used to gain access to user data after the TOE entered a Compliant power saving state. The TSS is inspected to ensure it describes that each authorization factor satisfies the requirements of FCS_AFA_EXT.1.1.

Guidance

The evaluator shall examine the guidance documentation for a description of authorization factors used to access plaintext data when resuming from a Compliant power saving state.

Test

The evaluator shall perform the following test:

- Enter the TOE into a Compliant power saving state
- Force the TOE to resume from a Compliant power saving state
- Release an invalid authorization factor and verify that access to decrypted plaintext data is denied
- Release a valid authorization factor and verify that access to decrypted plaintext data is granted.

2.2.1 TSS

[ST] Section 7.1 states that a password is the only authorization factor used to gain access to user data after the TOE enters a Compliant power saving state.

2.2.2 Guidance

[AGD] Section 7 states a password is required in both Mode 1 and Mode 6 in order to exit D3 and enter D0. Only D0 allows a user to access plaintext data.

2.2.3 KMD

None.

2.2.4 Testing

This test is satisfied by testing in Section 2.22.4

2.3 FCS_CKM.1(b) Cryptographic Key Generation (Symmetric Keys)

TSS

The evaluator shall review the TSS to determine that a symmetric key is supported by the product, that the TSS includes a description of the protection provided by the product for this key. The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE.

Guidance

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key size(s) for all uses specified by the AGD documentation and defined in this cPP.

KMD

If the TOE uses a symmetric key as part of the key chain, the KMD should detail how the symmetric key is used as part of the key chain.

2.3.1 TSS

[ST] Section 7.1.1 describes all of the symmetric keys that are generated by the TOE. The keys are protected using key wrap before storing them. [ST] Section shows that all keys are 256-bits.

2.3.2 Guidance

[AGD] Section 15 states all data is encrypted with AES-256 XTS and that there is no configuration options or support for different key sizes.

2.3.3 KMD

All keys on the TOE are symmetric. [KMD] Section 3, Table 4 describes the key chain for both modes.

2.3.4 Testing

None.

2.4 FCS_CKM.1(c) Cryptographic Key Generation (Data Encryption Key)

TSS

The evaluator shall examine the TSS to determine that it describes how the TOE obtains a DEK (either generating the DEK or receiving from the environment).

If the TOE generates a DEK, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. If the DEK is generated outside of the TOE, the evaluator checks to ensure that for each platform identified in the TOE the TSS, it describes the interface used by the TOE to invoke this functionality. The evaluator uses the description of the interface between the RBG and the TOE to determine that it requests a key greater than or equal to the required key sizes.

If the TOE received the DEK from outside the host platform, then the evaluator shall examine the TSS to determine that the DEK is sent wrapped using the appropriate encryption algorithm.

KMD

If the TOE received the DEK from outside the host platform, then the evaluator shall verify that the KMD describes how the TOE unwraps the DEK.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall configure the TOE to ensure the functionality of all selections.

2.4.1 TSS

[ST] Section 7.1.1 of the TSS describes how the TOE generates the DEK in mode 1 and how the TOE revives the DEK (wrapped using AES-256 in KW mode) from the end user in mode 6. The section for mode 1 that describes DEK generation states that the DRBG is used, the preceding paragraph of this section describes, in detail, how the DRBG functions.

2.4.2 Guidance

None.

2.4.3 KMD

The TOE receives the DEK from outside the host platform in mode 6. [KMD] Section 3, Table 4, describes that the BEV is used to unwrap the DEK using AES-KW-256.

2.4.4 Testing

The evaluator configured the TOE into each evaluated configuration (Modes 1 and 6). For Mode 1, the evaluator configured a password authorization factor with a TOE generated DEK. For Mode 6, the evaluator configured the TOE with a password authorization factor, an evaluator supplied KEK, and an evaluator supplied DEK.

2.5 FCS_CKM.4(a) Cryptographic Key Destruction (Power Management)

TSS

The evaluator shall verify the TSS provides a high level description of how keys stored in volatile memory are destroyed. The valuator to verify that TSS outlines:

- if and when the TSF or the Operational Environment is used to destroy keys from volatile memory;
- if and how memory locations for (temporary) keys are tracked;
- details of the interface used for key erasure when relying on the OE for memory clearing.

Guidance Documentation

The evaluator shall check the guidance documentation if the TOE depends on the Operational Environment for memory clearing and how that is achieved.

KMD

The evaluator shall check to ensure the KMD lists each type of key, its origin, possible memory locations in volatile memory.

2.5.1 TSS

[ST] Section 7.1.2 states keys stored in volatile memory are over-written by zeros followed by a read verify. This section also states the TOE destroys keys on loss of power.

2.5.2 Guidance

[AGD] Section 15, bullet point 8 states the Administrator and/or system designers shall implement application techniques, safeguards, and/or procedures to assure that power is removed from the TOE, state D3 (cold), when the host system is left unattended. On removal of power, the TOE purges the DEK key and enters a full-off state in less than 20 milliseconds.

2.5.3 KMD

[KMD] Section 6, Table 5a and Table 5c, list each type of key, its origin, and possible memory locations in volatile memory.

2.5.4 Testing

None.

2.6 FCS_CKM.4(b) Cryptographic Key Destruction (TOE-Controlled Hardware)

TSS & KMD

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

The evaluator shall check to ensure the TSS lists each type of key that is stored, and identifies the memory type where key material is stored. When listing the type of memory employed, the TSS will list each type of memory selected in the FCS_CKM.4.1 SFR, as well as any memory types that employ a different memory controller or storage algorithm. For example, if a TOE uses NOR flash and NAND flash, both types are to be listed.

The evaluator shall examine the TSS to ensure it describes the method that is used by the memory controller to write and read memory from each type of memory listed. The purpose here is to provide a description of how the memory controller works so one can determine exactly how keys are written to memory. The description would include how the data is written to and read from memory (e.g., block level, cell level), mechanisms for copies of the key that could potentially exist (e.g., a copy with parity bits, a copy without parity bits, any mechanisms that are used for redundancy).

The evaluator shall examine the TSS to ensure it describes the destruction procedure for each key that has been identified. If different types of memory are used to store the key(s), the evaluator shall check to ensure that the TSS identifies the destruction procedure for each memory type where keys are stored (e.g., key X stored in flash memory is destroyed by overwriting once with zeros, key X' stored in EEPROM is destroyed by a overwrite consisting of a pseudo random pattern – the EEPROM used in the TOE uses a wear-leveling scheme as described).

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.

Upon completion of the TSS examination, the evaluator understands how all the keys (and potential copies) are destroyed.

Guidance

There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information. The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, 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, it is assumed the drive supports 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. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

It is assumed that if a RAID array is being used, only set-ups that support TRIM are utilized. It is assumed if the drive is connected via PCI-Express, the operating system supports TRIM over that channel. It is assumed the drive is healthy and contains minimal corrupted data and will be end of life before a significant amount of damage to drive health occurs, it is assumed there is a risk small amounts of potentially recoverable data may remain in damaged areas of the drive.

Finally, it is assumed the keys are not stored using a method that would be inaccessible to TRIM, such as being contained in a file less than 982 bytes which would be completely contained in the master file table.

For destruction on wear-leveled memory, if a time period is required before is processed destruction the ST author shall provide an estimated range.

Test

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.

For destruction on wear-leveled memory, if a time period is required before is evaluator shall wait that amount of time after clearing the key in tests 2 and 3.

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

1. Record the value of the key in the TOE subject to clearing.
2. Cause the 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.
7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a miniscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

Test 2: Applied to each key held in non-volatile 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 non-volatile 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.
5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for test 1 above), and if a fragment is found in the repeated test then the test fails.

Test 3: Applied to each key held as non-volatile 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. Read the storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

2.6.1 TSS & KMD

[ST] Section 7.1.2, Table 8 and Table 9, describe key length, how the keys are initialized, used, where the key is stored, and how the key is destroyed. [ST] Section 7.1.2 states the TOE's non-volatile memory is Magnetic RAM and that the plaintext DEK, when unwrapped, resides in the TOE's NAND controller's RAM and wrapped keys reside in a separate NVRAM.

[ST] Section 7.1.2 states there are no states or circumstances that the TOE does not conform to the key destruction requirements.

[ST] Section 7.1.2 describes that the memory controller reads memory in 32-bit blocks. It also claims that drive does not maintain copies of the keys for redundancy.

[ST] Section 7.1.2 describes that the TOE uses FPGA block RAM and it is accessed by blocks of 32 bits. It also describes that the TOE NVRAM is also accessed by 32-bit blocks.

[ST] Section 7.1.2 describes that no copies of keys are created in memory.

[KMD] Tables 5b and 5d describe key destruction procedures for every identified key for the two modes of operation, listing when and how keys are destroyed for each possible key location.

2.6.2 Guidance

[ST] Section 7.1.2 states there are no circumstances that the TOE does not conform to the key destruction requirements. In [ST] Section 6.1.1.6, the ST author selected the TOE does not employ a wear-leveling algorithm.

2.6.3 Testing

Special debug version of the firmware was developed by Mercury to be used to perform this testing. The evaluator conducted the testing. Additionally, the evaluator conducted source code review for key destruction procedures to verify same procedures are being employed for key zeroization.

Test 1:

The evaluator loaded known keys into the TOE.

The evaluator used a debug version of the firmware to dump TOE volatile memory and searched the memory dump for a known key value. No matches were found.

Test 2:

The evaluator loaded known keys into the TOE.

The evaluator used a debug version of the firmware to dump TOE non-volatile memory and searched the memory dump for a known key value. Key data was found.

The evaluator initiated the secure erase procedure for the TOE.

The evaluator used a debug version of the firmware to dump TOE non-volatile memory and searched the memory dump for a known key value. Key data was not found.

Test 3.

The evaluator loaded known keys into the TOE.

The evaluator used a debug version of the firmware to dump TOE non-volatile memory and searched the memory dump for a known key value. Key data was found.

The evaluator initiated the secure erase procedure for the TOE.

The evaluator used a debug version of the firmware to dump TOE non-volatile memory and verified that the previous key position is overwritten by zeroes.

2.7 FCS_CKM.4(d) Cryptographic Key Destruction (Software TOE, 3rd Party Storage)

TSS & KMD

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

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

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

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

Guidance

There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information. The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, 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, it is assumed the drive supports 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. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

It is assumed that if a RAID array is being used, only set-ups that support TRIM are utilized. It is assumed if the drive is connected via PCI-Express, the operating system supports TRIM over that channel. It is assumed the drive is healthy and contains minimal corrupted data and will be end of life before a significant amount of damage to drive health occurs, it is assumed there is a risk small amounts of potentially recoverable data may remain in damaged areas of the drive.

Finally, it is assumed the keys are not stored using a method that would be inaccessible to TRIM, such as being contained in a file less than 982 bytes which would be completely contained in the master file table.

Test

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

1. Record the value of the key in the TOE subject to clearing.
2. Cause the 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.
7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a miniscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

The following tests apply only to selection a), since the TOE in this instance has more visibility into what is happening within the underlying platform (e.g., a logical view of the media). In selection b), the TOE has no visibility into the inner workings and completely relies on the underlying platform, so there is no reason to test the TOE beyond test 1.

For selection a), the following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern.

Test 2: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media (e.g., MBR file system):

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.
5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for Use Case 1 test 1 above), and if a fragment is found in the repeated test then the test fails.

Test 3: Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media:

1. Record the logical storage location of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

2.7.1 TSS & KMD

[ST] Section 7.1.2, Table 8 and Table 9, list where each key is stored and how it is stored. [ST] Section 7.2.1 states that the platform communicates with the TOE via the SATA interface. It also states that the TOE does not store keys outside of the TOE.

[ST] Section 7.2.1. states that the TOE does not use platform-provided storage for key storage and management.

2.7.2 Guidance

[ST] Section 7.1.2 states there are no circumstances that the TOE does not conform to the key destruction requirements. In [ST] Section 6.1.1.6, the ST author selected that the TOE does not employ a wear-leveling algorithm.

2.7.3 Testing

The TOE does not use third party storage to store key materials.

2.8 FCS_CKM_EXT.4(a) Cryptographic Key and Key Material Destruction (Destruction Timing)

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 then should be expected to be destroyed.

KMD

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

The evaluator shall verify the KMD includes a key lifecycle, that includes a description where key material reside, how the key material is used, how it is determined that keys and key material are no longer needed, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(a) for the destruction.

2.8.1 TSS

[ST] Section 7.1.2 states that keys are considered no longer needed once the DEK is successfully unwrapped and data is flowing through the encryption engine.

2.8.2 Guidance

None.

2.8.3 KMD

[KMD] Section 6, Tables 5a, 5b, 5c, and 5d state where key and key material reside and when the keys and key material are no longer needed. These tables also state the usage and how the keys are destroyed.

2.8.4 Testing

None.

2.9 FCS_CKM_EXT.4(b) Cryptographic Key and Key Material Destruction (Power Management) (EE)

TSS

The evaluator shall verify the TSS provides a description of what keys and key material are destroyed when entering any Compliant power saving state.

Operational Guidance

The evaluator shall validate that guidance documentation contains clear warnings and information on conditions in which the TOE may end up in a non-Compliant power saving state indistinguishable from a Compliant power saving state. In that case it must contain mitigation instructions on what to do in such scenarios.

KMD

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside.

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM_EXT.6 for the destruction.

2.9.1 TSS

[ST] Section 7.1.2 states that the plaintext KEK used to protect the BEV along with interim keys, like the Black key, and other key material stored in volatile memory are erased during transition to a Compliant power saving state.

2.9.2 Guidance

[AGD] Section 7 states the TOE supports no other power saving states, and includes safeguards for preventing TOE from entering other, non-compliant power states.

2.9.3 KMD

[KMD] Section 6, Tables 5a, 5b, 5c, and 5d state where key and key material reside and when the keys and key material are no longer needed. These tables also state the usage and how the keys are destroyed.

2.9.4 Testing

None.

2.10 FCS_CKM_EXT.4(b) Cryptographic Key and Key Material Destruction (Power Management) (AA)

TSS

The evaluator shall verify the TSS provides a description of what keys and key material are destroyed when entering any Compliant power saving state.

Guidance

The evaluator shall validate that guidance documentation contains clear warnings and information on conditions in which the TOE may end up in a non-Compliant power saving state indistinguishable from a Compliant power saving state. In that case it must contain mitigation instructions on what to do in such scenarios.

KMD

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside.

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(d) for the destruction.

2.10.1 TSS

[ST] Section 7.1.2 states that the plaintext KEK used to protect the BEV along with interim keys, like the Black key, and other key material stored in volatile memory are erased during transition to a Compliant power saving state.

2.10.2 Guidance

[AGD] Section 7 states the TOE supports no other power saving states, and includes safeguards for preventing TOE from entering other, non-compliant power states.

2.10.3 KMD

[KMD] Section 6, Tables 5a, 5b, 5c, and 5d state where key and key material reside and when the keys and key material are no longer needed. These tables also state the usage and how the keys are destroyed.

2.10.4 Testing

None.

2.11 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

TSS/KMD

The evaluator shall examine the TOE's keychain in the TSS/KMD and verify all keys subject to destruction are destroyed according to one of the specified methods.

2.11.1 TSS/KMD

[ST] Section 7.1.2, Table 8 and Table 9, describe how each of the keys are destroyed.

[KMD] Section 6, Table 5b and Table 5d describe how each of the keys are destroyed.

2.11.2 Guidance

None.

2.11.3 Testing

None.

2.12 FCS_COP.1(a) Cryptographic Operation (Signature Verification)

This requirement is used to verify digital signatures attached to updates from the TOE manufacturer before installing those updates on the TOE. Because this component is to be used in the update function, additional Evaluation Activities to those listed below are covered in other evaluation activities sections in this document. The following activities deal only with the implementation for the digital signature algorithm; the evaluator performs the testing appropriate for the algorithm(s) selected in the component.

Hash functions and/or random number generation required by these algorithms must be specified in the ST; therefore the Evaluation Activities associated with those functions are contained in the associated

Cryptographic Hashing and Random Bit Generation sections. Additionally, the only function required by the TOE is the verification of digital signatures. If the TOE generates digital signatures to support the implementation of any functionality required by this cPP, then the applicable evaluation and validation scheme must be consulted to determine the required evaluation activities.

TSS

The evaluator shall check the TSS to ensure that it describes the overall flow of the signature verification. This should at least include identification of the format and general location (e.g., "firmware on the hard drive device" rather than “memory location 0x00007A4B") of the data to be used in verifying the digital signature; how the data received from the operational environment are brought on to the device; and any processing that is performed that is not part of the digital signature algorithm (for instance, checking of certificate revocation lists).

Test

Each section below contains the tests the evaluators must perform for each type of digital signature scheme. Based on the assignments and selections in the requirement, the evaluators choose the specific activities that correspond to those selections.

It should be noted that for the schemes given below, there are no key generation/domain parameter generation testing requirements. This is because it is not anticipated that this functionality would be needed in the end device, since the functionality is limited to checking digital signatures in delivered updates. This means that the domain parameters should have already been generated and encapsulated in the hard drive firmware or on-board non-volatile storage. If key generation/domain parameter generation is required, the evaluation and validation scheme must be consulted to ensure the correct specification of the required evaluation activities and any additional components.

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

The following tests may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Verification Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

RSA Signature Algorithm Tests

Signature Verification Test

The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party’s authentic and unauthentic signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.

The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.

2.12.1 TSS

[ST] Section 7.3.1 describes the overall flow of the signature verification process. The data used to verify the digital signature is stored in FPGA block in RAM. There is no additional processing that is not part of the digital signature algorithm.

2.12.2 Guidance

None.

2.12.3 KMD

None.

2.12.4 Testing

[ST] Section 1.4.3.1 identifies ECDSA Cert. #883, satisfying the FCS_COP.1(1) testing requirements.

2.13 FCS_COP.1(b) Cryptographic Operation (Hash Algorithm)

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.

Guidance

The evaluator checks the operational guidance documents to determine that any system configuration necessary to enable required hash size functionality is provided.

Test

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 cPP.

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 ≤ i ≤ m. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 99*i, where 1 ≤ i ≤ 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 ≤ i ≤ m/8. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 8*99*i, where 1 ≤ i ≤ 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 NIST Secure Hash Algorithm Validation System (SHAVS) (https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Algorithm-ValidationProgram/documents/shs/SHAVS.pdf). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

2.13.1 TSS

[ST] Section 7.3.1 – Table 10 – FCS_COP.1(b) states that the module implements SHA-512 with a block size of 1024 which is used in the DRNG as well as used as the hashing function in the HMAC portion of the PBKDF and ECDSA signature verification.

2.13.2 Guidance

[ST] Section 7.1.3 states, “The module implements SHA-512 with a block size of 1024 which is used in the DRNG as well as used as the hashing function in the HMAC portion of the PBKDF and ECDSA signature verification.” The hash size is not configurable by the TOE, so no configuration is required.

2.13.3 KMD

None.

2.13.4 Testing

[ST] Section 1.4.3.1 identifies SHS Cert. #3291 satisfying the FCS_COP.1(b) testing requirements.

2.14 FCS_COP.1(c) Cryptographic Operation (Keyed Hash Algorithm)

TSS

If HMAC was selected:

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.

If CMAC was selected:

The evaluator shall examine the TSS to ensure that it specifies the following values used by the CMAC function: key length, block cipher used, block size (of the cipher), and output MAC length used.

Test

If HMAC was selected:

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 compared to the result of generating HMAC tags with the same key using a known good implementation.

If CMAC was selected:

For each of the supported parameter sets, the evaluator shall compose at least 15 sets of test data. Each set shall consist of a key and message data. The test data shall include messages of different lengths, some with partial blocks as the last block and some with full blocks as the last block. The test data keys shall include cases for which subkey K1 is generated both with and without using the irreducible

polynomial R_b, as well as cases for which subkey K2 is generated from K1 both with and without using the irreducible polynomial R_b. (The subkey generation and polynomial R_b are as defined in SP800-38E.) The evaluator shall have the TSF generate CMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating CMAC tags with the same key using a known good implementation.

2.14.1 TSS

[ST] Section 7.3.1 – Table 10 – FCS_COP.1(c) states that the keyed hashing algorithm is used in SP800-132 PBKDF with the following parameters: 32 byte key, SHA-512 hash, 128-bit block, 512-bit MAC.

2.14.2 Guidance

None.

2.14.3 KMD

None.

2.14.4 Testing

[ST] Section 1.4.3.1 identifies HMAC Cert. #2602 satisfying the FCS_COP.1(c) testing requirements.

2.15 FCS_COP.1(d) Cryptographic Operation (Key Wrapping)

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.

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.

2.15.1 TSS

[ST] Section 7.3.1 – Table 10 – FCS_COP.1(d) states that all AES-KW operations use a 256-bit key and KW mode as specified in ISO/IEC 18033-3 NIST SP 800-38F.

2.15.2 Guidance

None.

2.15.3 KMD

[KMD] Section 6, Table 5a and Table 5c state that the keys are wrapped using AES-KW-256. This section also states that the key wrap is compliant with SP 800-38F.

2.15.4 Testing

None.

2.16 FCS_COP.1(f) Cryptographic Operation (AES Data Encryption/Decryption) TSS

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

Guidance

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

Test

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

AES-CBC Tests

For the AES-CBC tests described below, the plaintext, ciphertext, and IV values shall consist of 128-bit blocks. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known-good implementation.

These tests are intended to be equivalent to those described in NIST's AES Algorithm Validation Suite (AESAVS) (http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf). Known answer values tailored to exercise the AES-CBC implementation can be obtained using NIST's CAVS Algorithm Validation Tool or from NIST's ACPV service for automated algorithm tests (acvp.nist.gov), when available. It is not recommended that evaluators use values obtained from static sources such as the example NIST's AES Known Answer Test Values from the AESAVS document, or use values not generated expressly to exercise the AES-CBC implementation.

AES-CBC Known Answer Tests

KAT-1 (GFSBox):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different plaintext values for each selected key size 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.

To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different ciphertext values for each selected key size and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using a key value of all zeros and an IV of all zeros.

KAT-2 (KeySBox):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size 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.

To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size and obtain the plaintext that results from AES-CBC decryption of an all-zeros ciphertext using the given key and an IV of all zeros.

KAT-3 (Variable Key):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of keys for each selected key size (as described below) and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using each key and an IV of all zeros.

Key i in each set shall have the leftmost i bits set to ones and the remaining bits to zeros, for values of i from 1 to the key size. The keys and corresponding ciphertext are listed in AESAVS, Appendix E.

To test the decrypt functionality of AES-CBC, the evaluator shall use the same keys as above to decrypt the ciphertext results from above. Each decryption should result in an all-zeros plaintext.

KAT-4 (Variable Text):

To test the encrypt functionality of AES-CBC, for each selected key size, the evaluator shall supply a set of 128-bit plaintext values (as described below) and obtain the ciphertext values that result from AES-CBC encryption of each plaintext value using a key of each size and IV consisting of all zeros.

Plaintext value i shall have the leftmost i bits set to ones and the remaining bits set to zeros, for values of i from 1 to 128. The plaintext values are listed in AESAVS, Appendix D.

To test the decrypt functionality of AES-CBC, for each selected key size, use the plaintext values from above as ciphertext input, and AES-CBC decrypt each ciphertext value using key of each size consisting of all zeros and an IV of all zeros.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a plaintext message of length i blocks, and encrypt the message using AESCBC. The resulting ciphertext values shall be compared to the results of encrypting the plaintext messages using a known good implementation.

The evaluator shall test the decrypt functionality by decrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a ciphertext message of length i blocks, and decrypt the message using AESCBC. The resulting plaintext values shall be compared to the results of decrypting the ciphertext messages using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality for each selected key size using 100 3-tuples of pseudo-random values for plaintext, IVs, and keys.

The evaluator shall supply a single 3-tuple of pseudo-random values for each selected key size. This 3-tuple of plaintext, IV, and key is provided as input to the below algorithm to generate the remaining 99 3-tuples, and to run each 3-tuple through 1000 iterations of AES-CBC encryption.

```
# Input: PT, IV, Key
Key[0] = Key
IV[0] = IV
PT[0] = PT

for i = 1 to 100 {

    Output Key[i], IV[i], PT[0]
    for j = 1 to 1000 {
    if j == 1 {

        CT[1] = AES-CBC-Encrypt(Key[i], IV[i], PT[1])
        PT[2] = IV[i]

    } else {

        CT[j] = AES-CBC-Encrypt(Key[i], PT[j])
        PT[j+1] = CT[j-1]

    }

}

    Output CT[1000]

        If KeySize == 128 { Key[i+1] = Key[i] xor CT[1000] }
        If KeySize == 256 { Key[i+1] = Key[i] xor ((CT[999] << 128) | CT[1000]) }

    IV[i+1] = CT[1000]
    PT[0] = CT[999]

}
```

The ciphertext computed in the 1000th iteration (CT[1000]) is the result for each of the 100 3-tuples for each selected key size. 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 above, 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 216 bits, whichever is smaller.

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.

2.16.1 TSS

[ST] Section 7.3.1 – Table 10 – FCS_COP.1(f) states that the user data is encrypted using XTS-AES with a 512 bit key.

2.16.2 Guidance

[AGD] Section 15 states the TOE encrypts all data with AES-256 XTS. There are no configuration options or support for different key sizes.

2.16.3 KMD

None.

2.16.4 Testing

[ST] Section 1.4.3.1 identifies AES Cert. #2802 satisfying the FCS_COP.1(f) testing requirements.

2.17 FCS_KYC_EXT.1 Key Chaining (Initiator)

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 AES128, and no fewer than 256 bits for products that support AES-256.

KMD

The evaluator shall examine the KMD describes a high level description of the key hierarchy for all authorizations methods selected in FCS_AFA_EXT.1 that are used to protect the BEV. 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 or key derivation methods that meet FCS_COP.1(d) and FCS_KDF_EXT.1.

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.

2.17.1 TSS

[ST] Section 7.1.1 describes that the BEV will be 256-bits for both mode 1 and mode 6.

2.17.2 Guidance

None.

2.17.3 KMD

[KMD] Section 3, Table 4 describes the key chain. [KMD] Section 6, Figures 2 and 3 show the flow of the keys through the AA and EE. The evaluator determined that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the strength of the BEV is maintained throughout the key chain. [KMD] Sections 3 and 6 detail how the keys are used and interact; the strengths of the keys are described in these sections.

2.17.4 Testing

None.

2.18 FCS_KYC_EXT.2 Key Chaining (Recipient)

KMD

The evaluator shall examine the KMD to ensure it describes a high level key hierarchy and details of the key chain. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap or key derivation methods that meet FCS_KDF_EXT.1, FCS_COP.1(d), FCS_COP.1(e), and/or FCS_COP.1(g).

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 knowledge of the BEV and the effective strength of the DEK is maintained throughout the Key Chain.

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

2.18.1 TSS

None.

2.18.2 Guidance

None.

2.18.3 KMD

[KMD] Section 3, Table 4 describes the key chain. [KMD] Section 6, Figures 2 and 3 show the flow of the keys through the AA and EE. The evaluator determined that at no point the chain could be broken without a cryptographic exhaust or knowledge of the BEV and the strength of the DEK is maintained throughout the key chain. [KMD] Sections 3 and 6 detail how the keys are used and interact; the strengths of the keys are described in these sections.

2.18.4 Testing

None.

2.19 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

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 evaluator also verifies that the TSS provides a description of how the password is conditioned and the evaluator ensures it satisfies the requirement.

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.

Test

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.

2.19.1 TSS

[ST] Section 7.1.1 states that the password length is limited to 64 characters. This section also states how the TOE conditions the password.

[ST] Section 7.1.1 states that any types of 8-bit characters can be used as a password. As the TOE accepts byte submask through ATA interface, no encoding is performed on accepted data.

[ST] Section 7.1 claims that the TOE does not enforce minimum password length and complexity requirements.

2.19.2 Guidance

None.

2.19.3 KMD

The [ST] states the BEV and intermediary keys are 256-bits. The [KMD] describes the BEV and intermediary keys as 256-bits. [ST] Section 7.1.1 describes the method by which the password is first encoded and then fed to the PBKDF. [KMD] Section 6 describes how the output of the PBKDF HMAC-SHA-512 is used to form the submask and the input into the function is the same length as the BEV.

[ST] Section 7.1.1 describes PBKDF settings for the operation. It describes that PBKDF input is no more and no less than 64 8-bit characters, producing 512 bits of input for PBKDF HMAC-SHA-512 function.

[ST] Section 7.1.1 states the TOE uses SP 800-132 compliant PBKDF using HMAC-SHA-512 over 1063 iterations.

[ST] Section 7.1.3 describes the TOE uses HMAC-SHA-512 with following parameters: 32 byte key, SHA-512 hash, 128-bit block, 512-bit MAC.

2.19.4 Testing

Test 1: The evaluator verified the TOE supports passwords/passphrases up to 64 characters.

Test 2: N/A: The TOE does not support password/passphrase lengths greater than 64.

Test 3: The evaluator verified the TOE supports all possible 8-bit values as valid characters for the password/passphrase.

2.20 FCS_RBG_EXT.1 Random Bit Generation

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.

Guidance

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, and provides information regarding how to instantiate/call the DRBG for RBG services needed in this cPP.

Test

The evaluator shall perform 15 trials for the RNG implementation. If the RNG 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 RNG are valid.

If the RNG 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 RNG 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.

2.20.1 TSS

[ST] Section 7.1.1 states the DEK is provided wrapped by the user when operating in mode 6. In mode 1 the DEK is generated using the DRBG.

2.20.2 Guidance

[ST] only references one DRBG mechanism. There is no need for an administrator to configure the TOE to use it.

2.20.3 KMD

None.

2.20.4 Testing

[ST] Section 1.4.3.1 identifies DRBG Cert. #1179 satisfying the FCS_RBG_EXT.1 testing requirements.

2.21 FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

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 or by the Operational Environment. If external function is used for this purpose, the TSS should include the specific API that is called with inputs.

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.

2.21.1 TSS

[ST] Section 7.1.1 states that salts and nonces come from the internal RNG for internally generated keys. This section also states that the XTS AES tweak value is set to the logical sector value for each sector being accessed. XTS mode does not create IVs.

2.21.2 Guidance

None.

2.21.3 KMD

None.

2.21.4 Testing

None.

2.22 FCS_VAL_EXT.1 Validation

TSS

The evaluator shall examine the TSS to determine which authorization factors support validation.

The evaluator shall examine the TSS to review a high-level description if multiple submasks are used within the TOE, how the submasks are validated (e.g., each submask validated before combining, once combined validation takes place).

The evaluator shall also examine the TSS to determine that a subset or all of the authorization factors identified in the SFR can be used to exit from a Compliant power saving state.

Guidance Documentation

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

(conditional) If the validation functionality is specified by the ST author, the evaluator shall examine the operational guidance to ensure that it states the values that the TOE uses for limits regarding validation attempts.

The evaluator shall verify that the guidance documentation states which authorization factors are allowed to exit a compliant power saving state.

KMD

The evaluator shall examine the KMD to verify that it described the method the TOE employs to limit the number of consecutively failed authorization attempts.

The evaluator shall examine the vendor's KMD to ensure it describes how validation is performed. The description of the validation process in the KMD provides detailed information how the TOE validates the BEV.

The KMD describes how the process works, such that it does not expose any material that might compromise the submask(s).

Tests

The evaluator shall perform the following tests:

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

Test 2(EE): The evaluator shall force the TOE to enter a Compliant power saving state, attempt to resume it from this state, and verify that only a valid authorization factor as defined by the guidance documentation is sufficient to allow the TOE to exit the Compliant power saving state.

Test 2(AA): For each validated authorization factor, ensure that when the user provides an incorrect authorization factor, the TOE prevents the BEV from being forwarded outside the TOE (e.g., to the EE).

2.22.1 TSS

[ST] Section 7.1.1 states that the BEV is validated in both modes as the Black DEK is unwrapped using the derived key.

2.22.2 Guidance

The validation functionality is not configurable. [ST] Section 6.1.1.21 specifies validation is performed using key wrap. This section also states the TOE will perform a key sanitization of the DEK upon a configurable number of consecutive failed validation attempts. [SCPG] Sections 2.2 and 2.3 describe how to set the validation attempts limit to 10. [SCPG] Section 4.4.1 states the TOE considers an invalid password, encryption key, or firmware file an authentication failure. [AGD] Section 7 states a password (and a BLACK DEK for mode 6) is required when entering the D0 state, which is the same as exiting the D3 state.

2.22.3 KMD

[KMD] Section 5 describes that the TOE employs a limit on the maximum authorization attempts by destroying stored keys when the limit is reached.

[KMD] Section 6 states, "The authentication will fail if using the password derived BEV (KEK) against the stored BLACK DEK, fails the AES unwrap."

2.22.4 Testing

Test 1 and 2(EE):
The evaluator configured the TOE to Mode 1 to limit the number of consecutive failed authorization attempts to 5. The evaluator entered an invalid password/passphrase 5 times and verified the TOE erased the DEK. The evaluator configured the TOE to Mode 6 to limit the number of consecutive failed authorization attempts to 10, powered off the drive, powered On the drive, and verified that drive was locked.

The evaluator entered an invalid password/passphrase 5 times followed by an invalid KEK 5 times, and verified the TOE erased the DEK. Erasure of the DEK is the expected behavior to limit the number of consecutive failed authorization attempts.

Test2(AA):
BEV never leaves the TOE boundary regardless of successful authorization.

2.23 FDP_DSK_EXT.1 Protection of Data on Disk

TSS

The evaluator shall examine the TSS to ensure that the description is comprehensive in how the data is written to the disk and the point at which the encryption function is applied. The TSS must make the case that standard methods of accessing the disk drive via the host platforms operating system will pass through these functions.

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

The evaluator shall verify the TSS in performing the evaluation activities for this requirement. The evaluator shall ensure the comprehensiveness of the description, confirms how the TOE writes the data to the disk drive, and the point at which it applies the encryption function.

The evaluator shall verify that the TSS describes the initialization of the TOE and the activities the TOE performs to ensure that it encrypts all the storage devices entirely when a user or administrator first provisions the TOE. The evaluator shall verify the TSS describes areas of the disk that it does not encrypt (e.g., portions associated with the Master Boot Records (MBRs), boot loaders, partition tables, etc.). If the TOE supports multiple disk encryptions, the evaluator shall examine the administration guidance to ensure the initialization procedure encrypts all storage devices on the platform.

Guidance Documentation

The evaluator shall review the AGD guidance to determine that it describes the initial steps needed to enable the FDE function, including any necessary preparatory steps. The guidance shall provide instructions that are sufficient, on all platforms, to ensure that all hard drive devices will be encrypted when encryption is enabled.

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 product, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and areas which are not encrypted (e.g. boot loaders, portions associated with the Master Boot Record (MBRs), 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 host 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 for all platforms to ensure that when the user enables encryption, the product encrypts all hard storage devices. The evaluator shall verify that the KMD describes the data flow from the device's host 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 Master Boot Record area).

The evaluator shall verify that the KMD provides a description of the platform's boot initialization, the encryption initialization process, and at what moment the product enables the encryption. The evaluator shall validate that the product does not allow for the transfer of user 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.

Tests

The evaluator shall perform the following tests:

Test 1: Write data to random locations, perform required actions and compare:

- Ensure TOE is initialized and, if hardware, encryption engine is ready;
- Provision TOE to encrypt the storage device. For SW Encryption products, or hybrid products use a known key and the developer tools.
- Determine a random character pattern of at least 64 KB;
- Retrieve information on what the device TOE's lowest and highest logical address is for which encryption is enabled.

Test 2: Write pattern to storage device in multiple locations:

- For HW Encryption, randomly select several logical address locations within the device's lowest to highest address range and write pattern to those addresses;
- ~~For SW Encryption, write the pattern using multiple files in multiple logical locations.~~

Test 3: Verify data is encrypted:

- For HW Encryption:
    - engage device's functionality for generating a new encryption key, thus performing an erase of the key per FCS_CKM.4(a);
    - Read from the same locations at which the data was written;
    - Compare the retrieved data to the written data and ensure they do not match
- ~~For SW Encryption, using developer tools;~~
    - ~~Review the encrypted storage device for the plaintext pattern at each location where the file was written and confirm plaintext pattern cannot be found.~~
    - ~~Using the known key, verify that each location where the file was written, the plaintext pattern can be correctly decrypted using the key.~~
    - ~~If available in the developer tools, verify there are no plaintext files present in the encrypted range.~~

2.23.1 TSS

[ST] Section 7.2.1 states that all data write operations are performed using the SATA interface and are routed through the encryption engine. There are no SATA data areas that are not encrypted. This section also describes how the TOE is initialized for both mode 1 and mode 6 in order to ensure that it is ready to encrypt all data for a user.

2.23.2 Guidance

[AGD] Sections 5, 18, 19, 20 and [SCPG] Sections 2.2, 2.3, and 4.1 describe configuration of the TOE to ensure that FDE functionality is enabled.

2.23.3 KMD

[KMD] Sections 1 through 6 describe the data encryptions engine, its components, and details about its implementation. [KMD] Section 3, Figure 1, is the detailed block diagram of the TOE. The block diagram includes the location of the data encryption engine within the data path. [KMD] Sections 2 and 6 describe the deployed scenarios of the TOE that will result in the encryption of data on the TOE. [KMD] Section 3, Figure 1, shows the data flow from the device's host interface to the device's persistent media that is storing the data.

[KMD] Section 3 describes conditions when the encryption engine is bypassed.

[KMD] Section 3 describes boot sequence and claims the TOE will not accept or provide data before the encryption engine is correctly initialized.

2.23.4 Testing

The evaluator performed the following test in each mode with each model of the TOE. The evaluator wrote a 64KB random pattern to address 0, max LBA (starting at max LBA minus 64KB), and approximately the middle LBA (max LBA divided by 2 and rounded down). The evaluator changed the cryptographic key and read the three addresses the random pattern was written to. The evaluator verified the random pattern was not read back.

2.24 FMT_MOF.1 Management of Functions Behavior

TSS

If support for Compliant power saving state(s) are claimed in the ST, the evaluator shall ensure the TSS describes how these are managed and shall ensure that TSS describes how only privileged users (administrators) are allowed to manage the states.

Guidance

The evaluator to check if guidance documentation describes which authorization factors are required to change Compliant power saving state behavior and properties.

Test

The evaluator shall perform the following tests:

Test 1: The evaluator presents a privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior and properties are allowed.

Test 2: The evaluator presents a non-privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior are not allowed.

2.24.1 TSS

[ST] Section 7.4.2 describes that the compliant power saving states are configured by administrators when putting the TOE in the Evaluated Configuration using the Guidance Documentation. This section also states that only administrators can configure these states.

2.24.2 Guidance

[AGD] Section 7 states the Crypto Officer must configure the TOE to disable non-compliant power saving states. [AGD] Section 24 describes that the Crypto Officer role is authenticated using a Configuration Password.

2.24.3 KMD

None.

2.24.4 Testing

The evaluator disabled “Intermediate power save mode” by providing correct authorization credentials. The evaluator presented incorrect authorization credentials when trying to change the settings and verified that this operation was not allowed.

2.25 FMT_SMF.1 Specification of Management Functions (EE)

TSS

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE changes the DEK.

If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE cryptographically erases the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: If additional management functions are claimed in the ST, the evaluator shall verify that the TSS describes those functions.

Guidance Documentation

If item a) is selected in FMT_SMF.1.1: The evaluator shall review the AGD guidance and shall determine that the instructions for changing a DEK exist. The instructions must cover all environments on which the TOE is claiming conformance, and include any preconditions that must exist in order to successfully generate or re-generate the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: Default Authorization Factors: It may be the case that the TOE arrives with default authorization factors in place. If it does, then the selection in item D must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist.

Disable Key Recovery: The guidance for disabling this capability shall be described in the AGD documentation.

KMD

If item d) is selected in FMT_SMF.1.1: If the TOE offers the functionality to import an encrypted DEK, the evaluator shall ensure the KMD describes how the TOE imports a wrapped DEK and performs the decryption of the wrapped DEK.

Tests

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to change and cryptographically erase the DEK (effectively removing the ability to retrieve previous user data).

If item c) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.

If item d) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.

2.25.1 TSS

[ST] Section 7.3.2 states that the TSF allows a user to change the DEK by cryptographically erasing the DEK and re-provisioning the TOE. This section also states that the TSF allows the user to cryptographically erase the DEK by issuing the zeroize command or failing an administrator-configurable

number of consecutive authentication attempts. [ST] Section 7.3.2 states that the limit of consecutive failed attempts can be configured by the administrator. [ST] Section 7.3.2 states that an administrator can configure which mode the TOE operates in. [ST] Section 7.3.2 states that a user can change the password used in the PBKDF. [ST] Section 7.3.2 states the user can initiate an update of the TOE firmware with a series of SATA “DOWNLOAD MICROCODE” commands.

2.25.2 Guidance

[AGD] Section 20 describes how to erase and change the DEK.

[AGD] Section 25 describes how to initiate the TOE update using the MDU utility and ATA commands.

[AGD] Section 21 contains a description of how to change User and Master Passwords.

[AGD] Sections 18 and 19 contain the configuration step of disabling “Intermediate Power Save mode” during the TOE secure configuration.

[AGD] Section 24 and [ST] Section 7.3.2 contain information that factory-installed configuration password is empty.

The TOE does not contain Key Recovery functionality.

2.25.3 KMD

[KMD] Section 6 describes that, in mode 6, the DEK is imported as the “BLACK DEK” which is wrapped using AES-KW-256. This section also describes how the DEK is decrypted after being imported.

2.25.4 Testing

The evaluator performed the following management functions in course of testing:

- Change the DEK
- Cryptographically erase the DEK
- Initiate TOE firmware updates
- Import a wrapped DEK
- Configure the failed authentication count limit
- Configure the operational mode of the module
- Change the password used to unwrap the DEK (Mode 1)
- Change the password used to unwrap the Black KEK (Mode 6)

2.26 FMT_SMF.1 Specification of Management Functions (AA)

TSS

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to change the DEK.

If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to cryptographically erase the DEK.

If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the methods by which users may change the set of all authorization factor values supported.

If item d) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: If power saving states can be managed, the evaluator shall ensure that the TSS describes how this is performed, including how the TOE supports disabling certain power

saving states if more than one are supported. If additional management functions are claimed in the ST, the evaluator shall ensure the TSS describes the additional functions.

Guidance

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how the functions for A and B can be initiated by the user.

If item c) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how selected authorization factor values are changed.

If item d) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: Default Authorization Factors: It may be the case that the TOE arrives with default authorization factors in place. If it does, then the selection in section E must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist.

Disable Key Recovery: The guidance for disabling this capability shall be described in the AGD documentation.

Power Saving: The guidance shall describe the power saving states that are supported by the TSF, how these states are applied, how to configure when these states are applied (if applicable), and how to enable/disable the use of specific power saving states (if applicable).

Test

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to forward a command to the EE to change and cryptographically erase the DEK. The actual testing of the cryptographic erase will take place in the EE.

If item c) is selected in FMT_SMF.1.1: The evaluator shall initialize the TOE such that it requires the user to input an authorization factor in order to access encrypted data.

> Test 1: The evaluator shall first provision user authorization factors, and then verify all authorization values supported allow the user access to the encrypted data. Then the evaluator shall exercise the management functions to change a user's authorization factor values to a new one. Then he or she will verify that the TOE denies access to the user's encrypted data when he or she uses the old or original authorization factor values to gain access.

If item d) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.

Test 2 (conditional): If the TOE provides default authorization factors, the evaluator shall change these factors in the course of taking ownership of the device as described in the operational guidance. The evaluator shall then confirm that the (old) authorization factors are no longer valid for data access.

Test 3 (conditional): If the TOE provides key recovery capability whose effects are visible at the TOE interface, then the evaluator shall devise a test that ensures that the key recovery capability has been or can be disabled following the guidance provided by the vendor.

Test 4 (conditional): If the TOE provides the ability to configure the power saving states that are entered by certain events, the evaluator shall devise a test that causes the TOE to enter a specific power saving state, configure the TSF so that this activity causes a different state to be entered, repeat the activity, and observe the new state is entered as configured.

Test 5 (conditional): If the TOE provides the ability to disable the use of one or more power saving states, the evaluator shall devise a test that enables all supported power saving states and demonstrates that the TOE can enter into each of these states. The evaluator shall then disable the supported power saving

states one by one, repeating the same set of actions that were performed at the start of the test, and observe each time that when a power saving state is configured to no longer be used, none of the behavior causes the disabled state to be entered.

2.26.1 TSS

[ST] Section 7.3.2 states that the TSF allows a user to change the DEK by cryptographically erasing the DEK and re-provisioning the TOE. This section also states that the TSF allows the user to cryptographically erase the DEK by issuing the zeroize command or failing an administrator-configurable number of consecutive authentication attempts. [ST] Section 7.4.2 states that the configuration of compliant power saving states is performed when the TOE is being put into the evaluated configuration. [ST] Section 7.3.2 includes information on the configuration of the limit of failed authentication attempts and how the counter persists over the changing of power saving states. [ST] Section 7.3.2 states the user can initiate an update of the TOE firmware with a series of SATA “DOWNLOAD MICROCODE” commands.

2.26.2 Guidance

[AGD] Section 20 describes how to erase and change the DEK.

[AGD] Section 25 describes how to initiate the TOE update using the MDU utility or ATA commands.

[SCPG] Section 4.2.2 describes how to set a User ATA Password.

[AGD] Section 21 contains description of how to change User and Master Passwords.

[AGD] Section 24 and [ST] Section 7.3.2 contain information that factory-installed configuration password is empty.

[AGD] Sections 18 and 19 contain the configuration step of disabling “Intermediate Power Save mode” during the TOE secure configuration.

[SCPG] Sections 2.2 and 2.3 contain initial configuration instructions including setting CONFIGURATION_FLAGS parameter that disables “Intermediate Power Save mode”.

The TOE does not contain Key Recovery functionality.

2.26.3 KMD

None.

2.26.4 Testing

The evaluator performed the following management functions in course of testing:

- Change the DEK
- Cryptographically erase the DEK
- Set user password
- Change user password
- Initiate TOE firmware updates
- Import a wrapped DEK
- Configure the failed authentication count limit
- Configure the operational mode of the module
- Disable power-saving states
- Change the password used to unwrap the DEK (Mode 1)

- Change the password used to unwrap the Black KEK (Mode 6)

2.27 FMT_SMR.1 Security Roles

TSS

There are no TSS evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

Guidance

There are no guidance evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

2.27.1 TSS

See FMT_MOF.1 and FMT_SMF.1.

2.27.2 Guidance

See FMT_MOF.1 and FMT_SMF.1.

2.27.3 KMD

None.

2.27.4 Testing

None.

2.28 FPT_KYP_EXT.1 Protection of Key and Key Material[1]

TSS

The evaluator shall examine the TSS and verify it identifies the methods used to protect keys stored in non-volatile memory.

KMD

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

2.28.1 TSS

[ST] Section 7.1.1 describes that keys and key material are protected using AES-256 in KW mode and are stored in non-volatile memory in wrapped form.

2.28.2 Guidance

None.

[1] TD0458 modifies the TSS and KMD Evaluation Activities

2.28.3 KMD

[KMD] Section 6, Tables 5a and 5c list where each key is stored when powered on and how it is protected when stored. All keys stored in non-volatile memory are stored wrapped, matching the selection in the [ST]. Plaintext keys are stored in non-volatile memory only if they are not part of the key chain.

2.28.4 Testing

None.

2.29 FPT_PWR_EXT.1 Power Saving States (EE)[2]

TSS

The evaluator shall validate the TSS contains a list of Compliant power saving states.

Guidance Documentation

The evaluator shall ensure that guidance documentation contains a list of Compliant power saving states. If additional power saving states are supported, then the evaluator shall validate that the guidance documentation states how the use of non-Compliant power saving states are disabled.

Tests

The evaluator shall confirm that for each listed Compliant state all key/key materials are removed from volatile memory by using the test indicated by the selection in FCS_CKM_EXT.6.

2.29.1 TSS

[ST] Section 7.4.2 lists that the TOE supports the Compliant power saving states at D0 and D3.

2.29.2 Guidance

[AGD] Sections 18 and 19 contain configuration the step of disabling “Intermediate Power Save mode” during the TOE secure configuration.

[SCPG] Sections 2.2 and 2.3 contain initial configuration instructions including setting CONFIGURATION_FLAGS parameter that disables “Intermediate Power Save mode”.

2.29.3 KMD

None.

2.29.4 Testing

Evaluator ensures power-off is the only compliant power saving state. All key material is removed from volatile memory when entering a Power-off state.

2.30 FPT_PWR_EXT.1 Power Saving States (AA)

TSS

The evaluator shall validate the TSS contains a list of Compliant power saving states.

Guidance

[2] TD0460 modifies the Guidance Evaluation Activity

The evaluator shall ensure that guidance documentation contains a list of Compliant power saving states. If additional power saving states are supported, then the evaluator shall validate that the guidance documentation states how non-Compliant power states are disabled

Test

The evaluator shall confirm that for each listed compliant state all key/key materials are removed from volatile memory by using the test defined in FCS_CKM.4(d).

2.30.1 TSS

[ST] Section 7.4.2 lists that the TOE supports the Compliant power saving states at D0 and D3.

2.30.2 Guidance

[AGD] Sections 18 and 19 contain the configuration step of disabling “Intermediate Power Save mode” during the TOE secure configuration.

[SCPG] Sections 2.2 and 2.3 contain initial configuration instructions including setting CONFIGURATION_FLAGS parameter that disables “Intermediate Power Save mode”.

2.30.3 KMD

None.

2.30.4 Testing

The evaluator ensured power-off is the only compliant power saving state. All key material is removed from volatile memory when entering a Power-off state.

2.31 FPT_PWR_EXT.2 Timing of Power Saving States (EE)

TSS

The evaluator shall validate that the TSS contains a list of conditions under which the TOE enters a Compliant power saving state.

Guidance Documentation

The evaluator shall check that the guidance contains a list of conditions under which the TOE enters a Compliant power saving state. Additionally, the evaluator shall verify that the guidance documentation provides information on how long it is expected to take for the TOE to fully transition into the Compliant power saving state (e.g. how many seconds for the volatile memory to be completely cleared).

Test

The evaluator shall trigger each condition in the list of identified conditions and ensure the TOE ends up in a Compliant power saving state by using the test indicated by the selection in FCS_CKM_EXT.6.

2.31.1 TSS

[ST] Section 7.4.2 states that the Compliant power saving states are entered upon requests from the user and system shut down.

2.31.2 Guidance

[AGD] Section 7 states when power is removed from the TOE, the TOE enters power state D3. [AGD] Section 16 states on removal of power, the TOE purges the DEK key and enters a full-off state in less than 20 milliseconds.

2.31.3 KMD

None.

2.31.4 Testing

The evaluator powered off the drive and performed a read without entering the authentication factors. The evaluator verified that the drive will not read or write user data.

2.32 FPT_PWR_EXT.2 Timing of Power Saving States (AA)

TSS

The evaluator shall validate that the TSS contains a list of conditions under which the TOE enters a Compliant power saving state.

Guidance

The evaluator shall check that the guidance contains a list of conditions under which the TOE enters a Compliant power saving state. Additionally, the evaluator shall verify that the guidance documentation states whether unexpected power-loss events may result in entry to a non-Compliant power saving state and, if that is the case, validate that the documentation contains information on mitigation measures.

Test

The evaluator shall trigger each condition in the list of identified conditions and ensure the TOE ends up in a compliant power saving state by running the test identified in FCS_CKM.4(d).

2.32.1 TSS

[ST] Section 7.4.2 states that the Compliant power saving states are entered upon requests from the user and system shut down.

2.32.2 Guidance

[AGD] Section 7 allows for D0 and D3 compliant power saving states. Power loss puts the TOE in a compliant power saving state.

2.32.3 KMD

None.

2.32.4 Testing

The evaluator powered off the drive and performed a read without entering the authentication factors. The evaluator verified that the drive will not read or write user data.

2.33 FPT_TST_EXT.1 TSF Testing

TSS

The evaluator shall verify that the TSS describes the known-answer self-tests for cryptographic functions.

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TOE and the method by which the TOE tests those functions. The evaluator shall verify that the TSS includes each of these functions, the method by which the TOE verifies the correct operation of the function. The evaluator shall verify that the TSF data are appropriate for TSF Testing. For example, more than blocks are tested for AES in CBC mode, output of AES in GCM mode is tested without truncation, or 512-bit key is used for testing HMAC-SHA-512.

If FCS_RBG_EXT.1 is implemented by the TOE and according to NIST SP 800-90, the evaluator shall verify that the TSS describes health tests that are consistent with section 11.3 of NIST SP 800-90.

If any FCS_COP functions are implemented by the TOE, the TSS shall describe the known-answer self-tests for those functions.

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe, for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall determine that all of the identified functions/components are adequately tested on startup.

2.33.1 TSS

[ST] Section 7.4.4 lists self-tests that are being performed by the TOE and a brief description of the tests including the cryptographic and non-cryptographic tests. [ST] Section 7.4.4 states that the DRBG health check is performed conditionally per 800-90A section 11.3. [ST] Section 7.4.4 lists which self-tests are performed on start up. This section also states the data used for testing for the Known Answer tests is compliant to FIPS 180-4 NIST Test Vectors for the applicable functions and therefore is appropriate for testing.

2.33.2 Guidance

None.

2.33.3 KMD

None.

2.33.4 Testing

None.

2.34 FPT_TUD_EXT.1 Trusted Update

TSS

The evaluator shall examine the TSS to ensure that it describes information stating that an authorized source signs TOE updates and will have an associated digital signature. The evaluator shall examine the TSS contains a definition of an authorized source along with a description of how the TOE uses public keys for the update verification mechanism in the Operational Environment. The evaluator ensures the TSS contains details on the protection and maintenance of the TOE update credentials.

If the Operational Environment performs the signature verification, then the evaluator shall examine the TSS to ensure it describes, for each platform identified in the ST, the interface(s) used by the TOE to invoke this cryptographic functionality.

Guidance

The evaluator ensures that the operational guidance describes how the TOE obtains vendor updates to the TOE; the processing associated with verifying the digital signature of the updates (as defined in FCS_COP.1(a)); and the actions that take place for successful and unsuccessful cases.

Test

The evaluators shall perform the following tests (if the TOE supports multiple signatures, each using a different hash algorithm, then the evaluator performs tests for different combinations of authentic and unauthentic digital signatures and hashes, as well as for digital signature alone):

Test 1: The evaluator performs the version verification activity to determine the current version of the TOE. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.

Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that an update successfully installs on the TOE. The evaluator shall perform a subset of other evaluation activity tests to demonstrate that the update functions as expected.

2.34.1 TSS

[ST] Section 7.4.3 states that the vendor maintains control of the ECDSA P-256 which is used to sign valid firmware updates. The public portion of the key is stored as part of the firmware which is integrity check at each boot. When a user initiates the firmware update, the host device will send the signed update to the TOE which will then check the signature. An invalid signature causes the update image to be deleted and an error to be returned.

2.34.2 Guidance

[AGD] Section 25 states updates can be obtained using a secure FTP login. The FTP site is managed by Mercury Systems. The TOE will verify the digital signature. The TOE performs signature verification before accepting new firmware.

[AGD] Section 25 states that if signature is not valid the TOE deletes the firmware image and returns the error.

[SCPG] Section 4.4.1 contains description of corresponding error code.

2.34.3 KMD

None.

2.34.4 Testing

The evaluator queried the TOE's firmware version and verified that the TOE correctly reports its firmware version. The tester attempted to install a firmware update with an invalid signature and verified the TOE rejected the update. The tester attempted to install a firmware update with a valid signature and verified the TOE successfully installed the update.

3 SAR Assurance Activities and Results

3.1 ASE: Security Target Evaluation

3.1.1 ASE_CCL.1.8C

The evaluator shall check that the statements of security problem definition in the PP and ST are identical.

Result:

[ST] Section 3 contains the Security Problem Definition.

Both [PP1] and [PP2] Section 3 describe the Security Problem Definition.

3.1.2 ASE_CCL.1.9C

The evaluator shall check that the statements of security objectives in the PP and ST are identical.

Result:

[ST] Section 4 contains the Security Objectives.

[PP1] and [PP2] Section 4 describes the Security Objectives.

3.1.3 ASE_CCL.1.10C

The evaluator shall check that the statements of security requirements in the ST include all the mandatory SFRs in the cPP, and all of the selection-based SFRs that are entailed by selections made in other SFRs (including any SFR iterations added in the ST). The evaluator shall check that if any other SFRs are present in the ST (apart from iterations of SFRs in the cPP) then these are taken only from the list of optional SFRs specified in the cPP (the cPP will not necessarily include optional SFRs, but may do so). If optional SFRs from the cPP are included in the ST then the evaluator shall check that any selection-based SFRs entailed by the optional SFRs adopted are also included in the ST.

Result:

[ST] Section 6 contains the Statement of Security Requirements.

[PP1] and [PP2] Section 5 contain the mandatory SFRs.

[PP1] and [PP2] Appendix B contain the selection-based SFRs.

[PP1] and [PP2] Appendix A contain the optional SFRs.

The evaluator determined that the Statement of Security Requirements in the [ST] contains all of the mandatory SFRs from [PP1] and [PP2].

The evaluator determined that the Statement of Security Requirements in the [ST] contains all necessary selection-based SFRs from [PP1] and [PP2].

The evaluator verified that all SFRs in the Statement of Security Requirements in the [ST] are mandatory SFRs, optional SFRs, or appropriate selection-based SFRs.

3.2 ADV: Development

3.2.1 ADV_FSP.1

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.

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

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

Result:

[ST] Section 1.3.4 identifies the TSFI as the Serial ATA revision 2.6 specification.

[ST] Section 7 states the purpose of each SFR-enforcing TSFI. The TSFI is implicitly categorized as SFR-enforcing.

[ST] Section 1.3.4 and Section 7 specify the method of use for each SFR-enforcing TSFI. [AGD] contains a description of commands, parameters and error messages, generated through the TSFI.

[ST] Section 1.3.4 and Section 7 specify all parameters for each SFR-enforcing TSFI.

[ST] Section 7 traces the SFRs to the corresponding TSFI. All of the SFRs trace to the SATA port.

3.3 AGD: Guidance Documents

3.3.1 Operational User Guidance (AGD_OPE.1)

The evaluator shall check the requirements below are met by the operational guidance. It should be noted that operational guidance may take the form of an “integrator’s guide”, where the TOE developer provides a description of the interface (e.g., commands that the Host Platform may invoke to configure a SED).

Operational guidance documentation shall be 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.

Operational guidance must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The contents of the operational guidance will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

- The operational guidance 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.
- The operational guidance shall describe how to configure the IT environments that are supported to shut down after an administratively defined period of inactivity.
- The operational guidance shall identify system “sleeping” states for all supported operating environments and for each environment, provide administrative guidance on how to disable the sleep state. As stated above, the TOE developer may be providing an integrator’s guide and “power states” may be an abstraction that SEDs provide at various levels – e.g., may simply provide a command that the Host Platform issues to manage the state of the device, and the Host Platform is responsible for providing a more sophisticated power management scheme.
- The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Result:

[ST] Section 1.4.1 indicates the [AGD] is part of the TOE.

[AGD] contains the operational guidance as well as the preparative procedures, so this work unit is covered under AGD_PRE.1-3-PP.

[AGD] Sections 5, 18, 19, 20 and [SCPG] Sections 2.2, 2.3, and 4.1 describe configuration of the TOE. Cryptographic engine is part of the TOE and is not configurable.

[AGD] Section 16 instructs the operator to configure the IT environment to remove power from the TOE when the host platform is left unattended.

[AGD] Sections 18 and 19 contain the configuration step of disabling “Intermediate Power Save mode” during the TOE secure configuration.

[SCPG] Sections 2.2 and 2.3 contain initial configuration instructions including setting CONFIGURATION_FLAGS parameter that disables “Intermediate Power Save mode”.

[AGD] Section 13 states the TOE supports CC compliant modes of Mode 1 and Mode 6, and no other key management modes were evaluated during the course of the CC-Evaluation.

3.3.2 Preparative Procedures (AGD_PRE.1)

The evaluator shall check the requirements below are met by the preparative procedures.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

Preparative procedures shall be 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.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

Preparative procedures must 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 itself).

Preparative procedures must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The preparative procedures must include

- instructions to successfully install the TSF in each Operational Environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability.

Result:

[AGD] Table 2 lists the additional support documentation and a description of the document.

[ST] Table 15 lists guidance documentation that is part of the TOE and is available to users.

[AGD] Section 14 states the TOE will function correctly in all products that include a SATA interface and are compliant to the SATA and ATA7 specification. [AGD] Section 15 lists the assumptions about the operational environment. The evaluator confirmed that these assumptions meet the requirements in [ST] Sections 1.3.4 and 4.1.

[AGD] Section 23 describes SATA connector requirements.

[ST] Section 1.4.1 describes the physical scope of the TOE. This section also identifies the models that the documentation applies to. The models do not require different operational environment requirements. The preparative procedures address all platforms claimed in the [ST].

[AGD] Section 3 describes the product configuration and deployment, Section 4 describes the usage scenarios, Sections 18 and 19 contain instructions on configuration, and Section 22 describes the installation of the TOE into a host system. [AGD] Section 18 and Section 29 describes the secure configuration of the TOE, Section 26 describes the physical security, and Section 22 describes the installation of the TOE as part of the larger operational environment. [AGD] Sections 16, 17, 18, 19, 20, 22, and 24 contain instructions for providing administrative capability.

[SCPG] Section 2 contains the secure configuration procedure through ATA commands sequence.

3.4 ATE: Tests

3.4.1 Independent Testing (ATE_IND.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 examine the TOE to determine that it has been installed properly and is in a known state.

The evaluator shall prepare a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities. While it is not necessary to have one test case per test listed in an Evaluation Activity, the evaluator must show in the test plan that each applicable testing requirement in the SFR-related Evaluation Activities is covered.

The test plan identifies the platforms to be tested, and for any platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition and configuration of each platform to be tested, and any setup actions that are necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of any cryptographic engine to be used (e.g. for cryptographic protocols being evaluated).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives, and the expected results.

The test report (which could just be an updated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure, so that a fix was then installed

and then a successful re-run of the test was carried out, then the report would show a "fail" result followed by a "pass" result (and the supporting details), and not just the "pass" result[3] .

Result:

The evaluator verified the TOE was in one of the two evaluated modes for each test by following the preparative procedures. Following the preparative procedures as described in [AGD] also enabled the evaluator to verify the TOE was in a known state.

The evaluator generated a test plan that performed the test subset specified by the supporting documents for the claimed protection profiles ([SD1] and [SD2]). The test plan specifies detailed/repeatable test steps as well as clear expected results. The test plan identifies the test environment and identifies special test tools, scripts, and firmware images necessary to perform testing. No cryptographic engine configuration is available for the TOE.

The evaluator recorded the actual results in the test report and verified the actual results were consistent with the expected results before passing each test.

3.5 AVA: Vulnerability Assessment

3.5.1 Vulnerability Survey (AVA_VAN.1)

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 3 above, the evaluator shall perform the following activities.~~

~~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.~~

The evaluator performs evaluation activities as specified in [CEM] and refined [SD1] and [SD2]. The evaluator formulates hypotheses in accordance with the process defined in Appendix A of [SD1] and [SD2]. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3 of [SD1] and [SD2]. The evaluator then performs vulnerability analysis in accordance with Appendix A.2 [SD1] and [SD2]. The results of the analysis shall be documented in the report according to Appendix A.3 [SD1] and [SD2].

Result:

[3] It is not necessary to capture failures that were due to errors on the part of the tester or test environment. The intention here is to make absolutely clear when a planned test resulted in a change being required to the originally specified test configuration in the test plan, to the evaluated configuration identified in the ST and operational guidance, or to the TOE itself.

The evaluator conducted a search in prescribed sources for prescribed keywords. Search results were screened and hits that bear no relation to the evaluated technology (specifically where module or product names were shared but related to another class of products) were removed from consideration. The remaining results were retained for further analysis.

The evaluator removed duplicated entries from previously identified lists of CVEs which resulted in a list of 55 entries, 12 of which have publication date more recent then the publication date of this cPP. Those are: CVE-2019-10099, CVE-2019-10636, CVE-2019-10705, CVE-2019-10706, CVE-2019-11686, CVE-2019-13178, CVE-2019-13179, CVE-2019-13466, CVE-2019-13603, CVE-2019-1586, CVE-2019-1589, CVE-2019-6481. Those constituted a public-sourced flaw hypotheses.

The evaluator then investigated public-sourced flaw hypotheses and iTC-sourced flaw hypotheses and disproved all of entries as not applicable to the TOE. The evaluator concluded that no known vulnerabilities exist for the TOE.

4 References

Abbr.	Name	Version	Date
[PP1]	collaborative Protection Profile for Full Drive Encryption - Encryption Engine	2.0 + Errata 20190201	February 1, 2019
[SD1]	Supporting Document Mandatory Technical Document Full Drive Encryption: Encryption Engine	2.0 + Errata 20190201	February 2019
[PP2]	collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition	2.0 + Errata 20190201	February 1, 2019
[SD2]	Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition	2.0 + Errata 20190201	February 2019
[ST]	Security Target for Mercury Systems ASURRE-Stor™ Solid State Self-Encrypting Drive	1.1	February 6, 2020
[AGD]	Mercury Systems ASURRE-Stor® ASD256/512, and ADR256/512 Solid State Self-Encrypting Drives Non-Proprietary Administrative Guidance	1.5.1	February 4, 2020
[SCPG]	SSD Secure Configuration Programmer's Guide ASURRE-Stor FIPS 140-2, CC (CSfC) 256 GB and 512 GB Solid State Drives	1.5.1.00	November 21, 2019
[KMD]	ASURRE-Stor ASD256/512 and ADR256/512 Solid State Self-Encrypting Drives Key Management Description (KMD)	1.5.1.00	December 18, 2019