Assurance Activities Report

for

Seagate® Secure NVMe Self-Encrypting Drives

Version 1.0

25 April 2024

Prepared by:

Leidos Inc.
https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

Phison Electronics Corporation
No.1, Qun-Yi Road, Jhunan, Miaoli County,
Taiwan 350, R.O.C.

The TOE Evaluation was Sponsored by:

Seagate Technology, LLC
47488 Kato Road
Fremont, CA 94538

Evaluation Personnel:

Anthony Apted
Pascal Patin

Common Criteria Version:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.

Common Evaluation Methodology Version:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017.

Protection Profile:

- *collaborative Protection Profile for Full Drive Encryption – Encryption Engine*, Version 2.0+Errata 20190201, 1 February 2019

Revision History

Version	Date	Description
0.1	3 November 2023	Initial draft
1.0	5 April 2024	Final draft

Contents

1 Introduction

This document presents results from performing assurance activities associated with the Seagate® Secure NVMe Self-Encrypting Drives evaluation. This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the following Protection Profile:

- collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0+Errata, 1 February 2019.

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant CAVP certification and not through performance of any testing as specified in the supporting documents.

1.1 Technical Decisions

This subsection lists the Technical Decisions that have been issued by NIAP against the claimed Protection Profile, along with rationale as to their applicability or otherwise to this evaluation.

TD0458: FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities

This TD has been applied to this evaluation.

TD0460: FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states

This TD has been applied to this evaluation.

TD0464: FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states

This TD has been applied to this evaluation.

TD0606: FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE

N/A – this TD provides a technical recommendation regarding evaluation Network Attached Storage (NAS) devices against the FDE EE and FDE AA cPPs, but the devices comprising the TOE are not NAS devices.

TD0766: FIT Technical Decision for FCS_CKM.4(d) Test Notes

N/A – the ST does not claim FCS_CKM.4(d).

TD0769: FIT Technical Decision for FPT_KYP_EXT.1.1

This TD has been applied to this evaluation.

1.2 SAR Evaluation

The following Security Assurance Requirements (SARs) were evaluated during the evaluation of the TOE:

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
ATE_IND.1	Pass
AVA_VAN.1	Pass

The evaluation work units are listed in the proprietary ETR. The evaluators note per the PP evaluation activities that many of the SARs were successfully evaluated through completion of the associated evaluation activities present in the Supporting Document for the claimed PP.

1.3 References

[CPP_FDE_EE_V2.0E]	*collaborative Protection Profile for Full Drive Encryption – Encryption Engine*, Version 2.0+Errata, 1 February 2019
[CPP_FDE_EE_SD_V2.0E]	Supporting Document – Mandatory Technical Document – Full Drive Encryption: Encryption Engine, Version 2.0+Errata 20190201, 1 February 2019
[ST]	Seagate® Secure NVMe Self-Encrypting Drives Security Target, Version 0.24, 7 March 2024
[KMD]	Seagate® Secure NVMe Self-Encrypting Drives Common Criteria Full Drive Encryption – Encryption Engine Key Management Description, Version 0.95, 7 March 2024
[CCCG]	Seagate® Secure NVMe Self-Encrypting Drive Common Criteria Configuration Guide, Version 1.2, 7 March 2024
[TR]	Seagate® Secure NVMe Self-Encrypting Drives Common Criteria Test Report and Procedures, Version 1.1, 25 April 2024.
[AVA]	Seagate® Secure NVMe Self-Encrypting Drive Vulnerability Assessment, Version 1.0, 25 April 2024

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are reproduced from [CPP_FDE_EE_SD_V2.0E]. NIAP Technical Decisions have been applied and are identified as appropriate.

2.1 Cryptographic Support (FCS)

The following table lists the cryptographic functions supported by the TOE and associated SFRs, the specific algorithms that are claimed for these functions, and the relevant CAVP certificate validation lists and certificate numbers for each.

Functions	Standards	Certificates
FCS_COP.1(a) Cryptographic Operation (Signature Verification)		
RSA (4096 bits)	Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS	A3307: RSA SigVer (FIPS186-4) A3308: RSA SigVer (FIPS186-4)
FCS_COP.1(b) Cryptographic Operation (Hash Algorithm)		
SHA-256 (digest size 256 bits) SHA-512 (digest sizes 512 bits)	ISO/IEC 10118-3:2004	A3307: SHA2-256 A3307: SHA2-512 A3308: SHA2-512
FCS_COP.1(c) Cryptographic Operation (Message Authentication)		
HMAC-SHA-256	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	A3307: HMAC-SHA2-256
FCS_COP.1(d) Cryptographic Operation (Key Wrapping)		
AES in KW mode (256 bits)	NIST SP 800-38F	A3307: AES-KW
AES in CBC mode (256 bits) (prerequisite to AES-KW)	FIPS PUB 197	A3307: AES-CBC
FCS_COP.1(f) Cryptographic Operation (AES Encryption/Decryption)		
AES in XTS mode (256 bits)	ISO /IEC 18033-3 IEEE 1619	A3307: AES-XTS Testing Revision 2.0
FCS_RBG_EXT.1 Random Bit Generation		
HMAC_DRBG (any): 256 bits entropy	NIST SP 800-90A	A3307: HMAC DRBG

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

2.1.1.1 TSS Activities

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.

Section 6.2.1 of [ST] ("Cryptographic Key Generation (FCS_CKM.1(b), FCS_CKM.1(c))") states the TOE generates symmetric cryptographic keys using the HMAC_DRBG deterministic random bit generation algorithm. Table 6 in [ST] ("Key Table") identifies the TOE generates the following keys in addition to the Data Encryption Keys (DEKs): Transfer Encryption Keys (TEKs); and Key Encryption Keys (KEKs). Section 6.2.1 of [ST] states the key size of all symmetric keys (other than DEKs) is 256 bits. The TOE uses key wrapping (specifically, AES in KW mode) to protect TEKs and KEKs.

2.1.1.2 Guidance Activities

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.

Section “CC Operational Guidance”, subsection “Cryptographic Symmetric Key Sizes and Key Generation” of [CCECG] states the TOE generates all AES keys needed for its operation. The length of AES keys is not configurable. Intermediate keys are always 256 bits long.

2.1.1.3 KMD Activities

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.

Section 2 of [KMD] (“Keys and Key Hierarchy”) describes the keys used by the TOE in the key chain from BEV to DEK. Section 2.3 of [KMD] (“Intermediate Keys”) describes two symmetric keys—the Transfer Encryption Key (TEK) and the Key Encryption Key (KEK)—used in the key chain. It states the TOE generates the TEK and the KEK using its HMAC_DRBG implementation. Section 2.5 of [KMD] (“Key Hierarchy”) describes how the TEK and KEK are used in the key chain. The TOE wraps the DEK with AES Key Wrapping, using the KEK as the wrapping key, and stores the wrapped DEK in non-volatile memory. The TOE wraps the KEK with AES Key Wrapping, using the TEK as the wrapping key, and stores the wrapped KEK in non-volatile memory. The TOE wraps the TEK with AES Key Wrapping, using the key derived from the BEV (using PKDF2) as the wrapping key, and stores the wrapped TEK in non-volatile memory.

2.1.1.4 Test Activities

There are no test evaluation activities for this SFR.

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

2.1.2.1 TSS Activities

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 u ses 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.

Section 6.2.1 of [ST] (“Cryptographic Key Generation (FCS_CKM.1(b), FCS_CKM.1(c))”) states the TOE uses its CAVP-validated implementation of HMAC_DRBG(any) to generate symmetric cryptographic keys. The specified symmetric cryptographic key size is always 512 bits for DEKs. AES in XTS-256 mode uses separate 256-bit keys for Initialization Vector (IV) and block encryption, which results in the need to provide 512 bits of key material for AES-XTS-256 mode.

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.

As specified in FCS_CKM.1(c) and described in Section 6.2.1 of [ST], the TOE generates its own DEKs and does not receive them from outside the host platform.

2.1.2.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.2.3 KMD Activities

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.

The TOE does not receive the DEK from outside the host platform.

2.1.2.4 Test Activities

The evaluator shall also perform the following tests:

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

The evaluator demonstrated the ability to call the TOE's DRBG functionality to generate a DEK.

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

2.1.3.1 TSS Activities

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.

Section 6.2.2 of [ST] ("Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), and FCS_CKM_EXT.6)") states the TOE destroys keys stored in volatile memory when power is removed, the drive is locked, or the TOE generates a new key to erase a locking range. When the TOE is powered off, all keys in volatile memory are destroyed. When the drive is Locked, all keys are overwritten with zeros. When the TOE generates a new key to erase a locking range, the existing key is overwritten with a new value of a key. Unlocked locking range keys are stored in plaintext form in volatile memory but as cyphertext in non-volatile memory for use by the encryption engine as needed. All other plaintext keys are temporarily stored in volatile memory on the stack for a short time after being generated and during the following operations: Take Ownership; Verify PIN. The keys are removed immediately after they are used or when they are no longer needed, using a single overwrite of zeroes.

2.1.3.2 Guidance Activities

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

The TOE does not depend on the Operational Environment for memory clearing.

2.1.3.3 KMD Activities

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

Section 2 of [KMD] (“Key and Key Hierarchy”) provides information for each of the keys required by the TOE, including for each key its type, origin, and possible memory locations in volatile memory.

2.1.3.4 Test Activities

There are no test evaluation activities for this SFR.

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

2.1.4.1 TSS + KMD Activities (KMD may be used if necessary details describe proprietary information)

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.

Section 2 of [KMD] (“Keys and Key Hierarchy”) provides the following information regarding management in volatile memory for each of the keys required by the TOE:

- Authentication PIN—the TOE receives the Authentication PIN as a BEV from the Authorization Acquisition function in the TOE’s operational environment. The TOE stores the Authentication PIN in plaintext in volatile memory.
- Password-based Key—the TOE derives the Password-based Key from the Authentication PIN using PBKDFv2 with a randomly-generated 256-bit salt and the HMAC-SHA-256 keyed hash algorithm. The TOE stores the Password-based key in plaintext in volatile memory.
- TEK—the TOE generates the TEK using HMAC_DRBG and wraps it using AES Key Wrapping with the Password-based Key before storing it in non-volatile memory. The TOE unwraps the wrapped TEK using AES Key Wrapping with the Password-based Key and stores it in volatile memory when it is needed to wrap and unwrap the KEK.
- KEK—the TOE generates the KEK using HMAC_DRBG and wraps it using AES Key Wrapping with the TEK before storing it in non-volatile memory. The TOE unwraps the wrapped KEK using AES Key Wrapping with the TEK and stores it in volatile memory when it is needed to wrap and unwrap the DEK.
- DEK—the TOE generates the DEK using HMAC_DRBG and wraps it using AES Key Wrapping with the KEK before storing it in non-volatile memory. The TOE unwraps the wrapped DEK using AES Key Wrapping with the KEK and stores it in volatile memory when it is needed to encrypt and decrypt user data.

Section 6.2.2 of [ST] (“6.2.2 Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), and FCS_CKM_EXT.6)”) states the TOE destroys keys stored in volatile memory in the following manner:

- When the TOE is powered off, all keys stored in volatile memory are destroyed.
- When the drive is locked, the TOE overwrites all keys with zeroes.
- When the TOE generates a new key to erase a locking range, the existing key is overwritten with the new value of the key.

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.

Section 3.1 of [KMD] ("Components") states the TOE includes volatile memory, in the form of Dynamic RAM (DRAM), and non-volatile memory in the form of NAND flash. Both memory types are accessed using standard microcontroller memory interface controllers and addressing schemes. DRAM is bit-level addressable while the NAND flash is block-level read and writeable.

Section 2 of [KMD] identifies the following keys that are stored in volatile memory (DRAM):

- Authentication PIN—the TOE stores the Authentication PIN in plaintext in volatile memory.
- Password-based Key—the TOE stores the Password-based key in plaintext in volatile memory.
- TEK—the TOE stores the TEK in plaintext in volatile memory when it is needed to wrap and unwrap the KEK.
- KEK— the TOE stores the KEK in plaintext in volatile memory when it is needed to wrap and unwrap the DEK.
- DEK—the TOE stores the DEK in plaintext in volatile memory when it is needed to encrypt and decrypt user data.

Section 2 of [KMD] identifies the following keys that are stored in non-volatile memory (NAND flash):

- TEK—the TOE wraps the TEK using AES Key Wrapping with the Password-based Key before storing it in non-volatile memory.
- KEK—the TOE wraps the KEK using AES Key Wrapping with the TEK before storing it in non-volatile memory.
- DEK—the TOE wraps the DEK using AES Key Wrapping with the KEK before storing it in non-volatile memory.

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).

Section 3.1 of [KMD] ("Components") provides the following information about the TOE. The physical boundary of the TOE is the physical drive enclosure. The physical interface to the TOE is the PCIe connector and jumper block pins. The logical interface is the industry standard NVMe and Opal SSC protocols, carried on the PCIe interface. The TOE contains a single Phison PS5020-E20 ASIC SSD Controller that includes a built-in hardware AES encryption engine. The PS5020-E20 directly controls the TOE's memory components. The TOE supports volatile DRAM and non-volatile NAND flash memory. In both cases, the TOE accesses memory using standard microcontroller memory interface controllers and addressing schemes. The DRAM is bit-level addressable and the NAND flash is block-level readable and writable.

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).

Section 6.2.2 of [ST] describes the destruction procedure for all keys when they are stored in volatile memory (DRAM) and when they are stored in non-volatile memory (NAND flash).

The TOE destroys keys stored in volatile memory in the following manner:

- When the TOE is powered off, all keys stored in volatile memory are destroyed.
- When the drive is locked, the TOE overwrites all keys with zeroes.
- When the TOE generates a new key to erase a locking range, the existing key is overwritten with the new value of the key.

The TOE destroys all keys stored in non-volatile memory by overwriting with a new value of the key. The non-volatile memory system performs any erase or wear leveling functions as necessary.

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 ST does not make use of the open assignment.

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

The evaluator checked Section 6 of [ST] ("TOE Summary Specification") and did not find any configurations or circumstances that did not conform strictly to key destruction requirement.

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

The successful completion of the other TSS evaluation activities in this section demonstrates the evaluator's understanding of how all keys (and potential copies) are destroyed by the TOE.

2.1.4.2 Guidance Activities

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.

Section “CC Operational Guidance”, subsection “Cryptographic Key Destruction” of [CCCG] states the TOE implements a NAND flash wear leveling algorithm. However, this algorithm does affect keys stored in NAND flash. The TOE does not store keys in NAND flash in plaintext—all keys are wrapped using 256-bit AES key wrapping. The TOE destroys all such keys by overwriting them with new values of the keys, thereby directly erasing the old (wrapped) key value.

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

As noted in section “CC Operational Guidance”, subsection “Cryptographic Key Destruction” of [CCCG], the wear leveling algorithm does not affect destruction of keys in NAND flash, so there is no time period required before keys are destroyed.

2.1.4.3 Test Activities

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.

The evaluator wrote data to the drive as per standard procedure. The TOE's volatile memory was then dumped to a file to verify that the unencrypted DEK and KEK exist in memory. The TCG Revert command was then executed to change or zeroize keys. Another memory dump was then performed. The contents of that dump was searched for both the entire DEK and KEK as well as fragments of those keys. Neither were found.

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.

The evaluator wrote data to the drive as per standard procedure. The TOE's non-volatile memory was then dumped to a file to verify that the encrypted DEK and KEK exist in memory. The TCG Revert command was then executed to change or zeroize keys. Another memory dump was then performed. The contents of that dump was searched for both the entire encrypted DEK and KEK as well as fragments of those keys. Neither were found.

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.

The evaluator wrote data to the drive as per standard procedure. The TOE's volatile memory was then dumped to a file to verify that the unencrypted DEK and KEK exist in memory. The TCG Revert command was then executed to change or zeroize keys. Another memory dump was then performed. The storage

location of the old DEK and KEK were examined, and found to contain a new key and zeroized data respectively.

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

2.1.5.1 TSS Activities

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.

Section 6.2.2 of [ST] ("Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), and FCS_CKM_EXT.6)") states the TOE stores keys in the key hierarchy of an unlocked locking range unencrypted in volatile memory (these keys are always wrapped when in non-volatile memory). All other plaintext keys are stored temporarily in volatile memory and on the stack for a short time after being generated and during specific operations (Take Ownership, Verify PIN). The TOE destroys all plaintext keys in volatile memory immediately after they are used or when they are no longer needed, using a single overwrite of zeroes.

2.1.5.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.5.3 KMD Activities

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.

Section 3.3 of [KMD] ("Operation") states the TOE supports three categories of storage: unencrypted for OS use; unencrypted for drive use; and encrypted for user data. The "unencrypted for OS use" storage includes the shadow MBR used for boot. The "unencrypted for drive use" storage is also termed the system area. The host OS has no access to the system area. The TOE uses the system area to store TCG Data Store tables and wrapped keys.

Section 2.5 of [KMD] ("Key Hierarchy") states the TOE does not store plaintext keys persistently and destroys all keys and keying material when no longer needed. When a drive range is locked, the plaintext value of the associated DEK for that range is erased from volatile memory using a single overwrite of zeroes.

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.

Section 2 of [KMD] ("Keys and Key Hierarchy") provides the key lifecycle, including descriptions of where key material resides and how it is used. Section 2.5 describes how it is determined keys and key material are no longer needed and how they are destroyed. The descriptions are consistent with the requirements in FCS_CKM.4(a).

2.1.5.4 Test Activities

There are no test evaluation activities for this SFR.

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

2.1.6.1 TSS Activities

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

Section 6.2.2 of [ST] ("Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), and FCS_CKM_EXT.6)") states the TOE destroys all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state (device full off (D3)).

2.1.6.2 Guidance Activities

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.

Section "CC Operational Guidance", subsection "Power Saving States and Timing of Power Saving States" states the TOE supports a single Compliant power state of device full off (D3). The TOE has two possible power state transitions: power off to power on; and power on to power off. It is not possible for the TOE to end up in a non-Compliant power saving state that is indistinguishable from the Compliant power saving state.

2.1.6.3 KMD Activities

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

Section 3.3 of [KMD] ("Operation") states the TOE supports three categories of storage: unencrypted for OS use; unencrypted for drive use; and encrypted for user data. The "unencrypted for OS use" storage includes the shadow MBR used for boot. The "unencrypted for drive use" storage is also termed the system area. The host OS has no access to the system area. The TOE uses the system area to store TCG Data Store tables and wrapped keys.

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.

Section 2 of [KMD] ("Keys and Key Hierarchy") provides the key lifecycle, including descriptions of where key material resides and how it is used. Section 2.5 describes how it is determined keys and key material are no longer needed and how they are destroyed. The documentation in the KMD follows FCS_CKM_EXT.6 (and, by extension, FCS_CKM.4(b)) for key destruction.

2.1.6.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.7 Cryptographic Key Destruction Types (FCS_CKM_EXT.6)

2.1.7.1 TSS + KMD Activities (KMD may be used if necessary details describe proprietary information)

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.

Section 5.2.1.7 of [ST] ("Cryptographic Key Destruction Types (FCS_CKM_EXT.6)") specifies the TSF shall use the key destruction methods specified in FCS_CKM.4(b), which are as follows:

- For volatile memory, destruction is accomplished using one of the following methods: a single overwrite consisting of zeroes; a single overwrite consisting of a new value of the key; removal of power to memory.
- For non-volatile memory that employs a wear-leveling mechanism, destruction is accomplished by using one of the following methods: overwrite with a new value of a key of the same size; block erase.

Section 4.3 of [KMD] ("Key Destruction") describes the following approaches for destroying keys stored in volatile memory and keys stored in non-volatile memory:

- Key stored in volatile memory:
  - All keys are destroyed when power is removed from the drive.
  - DEKs are overwritten with a new value of the key.
  - TEKs and KEKs are overwritten with zeroes.
- Key stored in non-volatile memory:
  - DEKs are overwritten with a new wrapped value of the key
  - TEKs and KEKs are overwritten with zeroes.

The appropriate key destruction method is applied to all keys in the TOE's keychain subject to destruction, depending on whether the key is in volatile or non-volatile memory.

2.1.7.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.7.3 Test Activities

There are no test evaluation activities for this SFR.

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

2.1.8.1 TSS Activities

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).

Section 6.2.3 of [ST] ("Cryptographic Operation (FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f))") states the TOE performs RSA Digital Signature Algorithm signature verification using a key size (modulus) of 4096 bits to verify TOE firmware updates.

Section 6.5.1 of [ST] ("Firmware Access Control and Update Authentication (FPT_FAC_EXT.1, FPT_FUA_EXT.1) describes the process the administrator follows to perform a firmware update. The administrator downloads the firmware update package from the vendor web site to the TOE. The currently executing TOE firmware receives the firmware update package it and writes it to volatile memory. The TOE uses the RSA Digital Signature Algorithm with the Root of Trust for Update (RTU) public key stored in ROM to verify the digital signature on the firmware update package.

2.1.8.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.8.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.8.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certification verifying RSA signature verification, as follows.

Algorithm	Tested Capabilities	Certificates
RSA as defined in FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS	Signature Verification Signature Type: PKCSPSS Modulo: 4096 Hash Algorithm: SHA2-512	A3307: RSA SigVer (FIPS186-4) A3308: RSA SigVer (FIPS186-4)

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

2.1.9.1 TSS Activities

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.

Section 6.2.3 of [ST] ("Cryptographic Operation (FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f))") states the TOE uses SHA-256 cryptographic hashing with HMAC-SHA-256 message authentication and HMAC_DRBG functions, and uses SHA-512 cryptographic hashing as part of RSA signature verification.

2.1.9.2 Guidance Activities

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

Section "CC Operational Guidance," subsection "Cryptographic Operation (Hash Algorithm)" of [CCCG] states no system configuration is necessary to enable the required hash size functionality.

2.1.9.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.9.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certification verifying cryptographic hashing functions, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in ISO/IEC 10118-3:2004	SHA-256 SHA-512	A3307: SHA2-256 A3307: SHA2-512 A3308: SHA2-512

2.1.10 Cryptographic Operation (Message Authentication) (FCS_COP.1(c))

2.1.10.1 TSS Activities

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.

Section 6.2.3 of [ST] ("Cryptographic Operation (FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f))") specifies the following values used with the TOE's HMAC function: key length of 256 bits; SHA-256 hash function; block size of 64 bytes (512 bits); and output MAC length of 32 bytes (256 bits).

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.

The ST does not select CMAC in FCS_COP.1(c), so this activity is not applicable.

2.1.10.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.10.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.10.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certification verifying HMAC keyed hash algorithm, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC that meets : ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	HMAC-SHA2-256	A3307: HMAC-SHA2-256

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

2.1.11.1 TSS Activities

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.

Section 6.2.3 of [ST] ("Cryptographic Operation (FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f))") states the TOE performs AES Key Wrap and AES Key Unwrap in accordance with NIST SP 800-38F. The TOE uses AES Key Wrap to protect intermediate keys in the key hierarchy and the Data Encryption Keys.

2.1.11.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.11.3 KMD Activities

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.

Section 2.5 of [KMD] ("Key Hierarchy") identifies three keys in the key hierarchy that are wrapped: Transfer Encryption Key (TEK); Key Encryption Key (KEK); and Data Encryption Key (DEK).

The TOE wraps the TEK using AES Key Wrap as defined in NIST SP 800-38F, using the password-based key derived from the Authentication Key as the encryption key. The TOE stores the wrapped TEK in non-volatile (NAND flash) memory.

The TOE wraps the KEK using AES Key Wrap as defined in NIST SP 800-38F, using the TEK as the encryption key. The TOE stores the wrapped KEK in non-volatile (NAND flash) memory.

The TOE wraps the DEK using AES Key Wrap as defined in NIST SP 800-38F, using the KEK as the encryption key. The TOE stores the wrapped DEK in non-volatile (NAND flash) memory.

Table 5 of [KMD] ("Key Wrapping") identifies when key wrapping of each key occurs.

2.1.11.4 Test Activities

There are no test evaluation activities for this SFR.

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

2.1.12.1 TSS Activities

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

Section 6.2.3 of [ST] ("Cryptographic Operation (FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f))") states the TOE performs AES XTS mode encryption using a key size of 256 bits.

2.1.12.2 Guidance Activities

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.

In the context of FCS_COP.1(f), the TOE supports a single mode of encryption (XTS). Therefore, this activity is not applicable.

2.1.12.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.12.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certification verifying AES data encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES-XTS as defined in IEEE 1619	Direction: Decrypt, Encrypt Key Length: 256 Payload Length: 128-4096 Increment 128 Tweak Mode: Number Data Unit Length Matches Payload	A3307: AES-XTS Testing Revision 2.0

2.1.13 Cryptographic Key Derivation (FCS_KDF_EXT.1)

2.1.13.1 TSS Activities

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

Section 6.2.4 of [ST] ("Cryptographic Key Derivation (FCS_KDF_EXT.1)") states the TOE implements PBKDFv2 with HMAC-SHA-256 and a randomly generated 256-bit salt value to transform the Authentication PIN into a derived key as specified in SP 800-132.

2.1.13.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.13.3 KMD Activities

The evaluator shall examine the vendor's KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived.

Section 2.5 of [KMD] ("Key Hierarchy") describes the source of each of the keys and critical security parameters in the key hierarchy. Only the password-based key is derived from other material—all other keys and critical security parameters (except for the Authentication PIN entered by the user) are generated by the TOE using its approved HMAC_DRBG function. The TOE derives the password-based key using PBKDFv2 as specified in NIST SP 800-132, using HMAC-SHA-256 as the pseudo- random function with a 256-bit random salt.

2.1.13.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.14 Key Chaining (Recipient) (FCS_KYC_EXT.2)

2.1.14.1 TSS Activities

There are no TSS evaluation activities for this SFR.

2.1.14.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.14.3 KMD Activities

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).

Section 2 of [KMD] ("Keys and Key Hierarchy") describes the high-level key hierarchy and provides a detailed description of the key chain.

The TOE supports up to four Locking SP Admin passwords or PINs and up to nine User passwords or PINs. The TOE receives a 32-byte (256-bit) authentication PIN from the host Authorization Acquisition component. The TOE validates the PIN by first calling the PBKDF with the PIN and associated plaintext salt value as inputs. The output of the PBKDF is the ephemeral plaintext password-based key associated with the PIN (FCS_KDF_EXT.1).

The TOE uses its approved HMAC_DRBG function to generate a 256-bit random number called the Transfer Encryption Key (TEK). The TOE uses the AES Key Wrap function to wrap the TEK using the password-based key (FCS_COP.1(d)).

The TOE uses its approved HMAC_DRBG function to generate a 256 bit random number called the Key Encryption Key (KEK). The TOE uses the AES Key Wrap function to wrap the KEK using the TEK (FCS_COP.1(d)).

The TOE uses its approved HMAC_DRBG function to generate two 256 bit random numbers that it uses to form the 512 bit Date Encryption Key (DEK). This is an AES-XTS key with a 256 bit encryption/decryption key and a 256 bit "tweak" key, as defined in IEEE 1619. The TOE uses the AES Key Wrap function to wrap the DEK using the KEK (FCS_COP.1(d)).

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.

Section 2.5 of [KMD] ("Key Hierarchy") describes how the key chain process functions. The key chain process does not expose any material that might compromise a key in the chain. When keys are

generated, they either are stored in volatile memory or they are wrapped before being stored in non-volatile memory.

Section 2.5 of [KMD] includes Figure 1 (“Key Hierarchy Diagram”), which illustrates the key hierarchy implemented by the TOE. Section 2.3 of [KMD] (“Intermediate Keys”) includes a table that details where the intermediate keys and keying material in the key hierarchy are stored and how each key originates or is derived. Section 2.4 of [KMD] describes the DEK, including how it is generated, hoe it is protected, and how it is stored.

The evaluator’s examination of the key hierarchy determined that at no point can the chain be broken without a cryptographic exhaust or knowledge of the BEV. The effective strength of the DEK of 256 bits is maintained throughout the key hierarchy.

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

The information in Section 2 of [KMD] describes the strength of every key and piece of keying material used throughout the key chain.

2.1.14.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.15 Random Bit Generation (FCS_RBG_EXT.1)

2.1.15.1 TSS Activities

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.

Section 6.2.6 of [ST] (“Random Bit Generation (FCS_RBG_EXT.1)”) states the TOE performs all deterministic random bit generation using HMAC_DRBG (any) with SHA-256 cryptographic hashing, in accordance with NIST SP 800-90A. The TOE does not rely on third party RBG services.

2.1.15.2 Guidance Activities

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.

Section “CC Operational Guidance”, subsection “Cryptographic Operation (Random Bit Generation)” of [CCECG] states the TOE automatically instantiates a NIST SP 800-90A compliant DRBG that is seeded with 256 bits of entropy from a NIST SP 800-90B compliant entropy system. Neither the DRBG nor the entropy system are configurable.

2.1.15.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.15.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 6.2 of [ST] ("Cryptographic Support"), Table 5 ("Cryptographic Functions") identifies the CAVP certification verifying random bit generation, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC_DRBG (any)	Mode: SHA2-256 Entropy Input: 256 Nonce: 128 Personalization String Length: 256 Additional Input: 256 Returned Bits: 256	A3307: HMAC DRBG

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

2.1.16.1 TSS Activities

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.

Section 6.2.7 of [ST] ("Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation (FCS_SNI_EXT.1)") describes how the TOE generates salts. It states the TOE uses 256 bit salt values generated randomly using the DRBG specified in FCS_RBG_EXT.1 (i.e., the HMAC_DRBG(any) algorithm).

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.

Section 6.2.7 of [ST] states the TOE uses tweak values with AES in XTS mode that are non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer. It further states the TOE does not use nonces or IV values.

2.1.16.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.16.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.16.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.17 Validation (FCS_VAL_EXT.1)

2.1.17.1 TSS Activities

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

Section 6.2.8 of [ST] ("Validation (FCS_VAL_EXT.1)") states the TOE validates each of the PINs used as authorization factors. The TOE validates the entered PIN by first calling the PBKDF with the PIN and the associated plaintext salt value as inputs. The output of the PBKDF is the ephemeral plaintext Authentication Key associated with the PIN. The TOE then calls the AES key unwrap function with the Authentication Key to unwrap the Transport Encryption Key (TEK) associated with the PIN. If the unwrap function produces the correct integrity check value (ICV), the PIN is valid, and authentication is successful. Otherwise, the PIN is invalid, and authentication is unsuccessful.

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 TOE does not use multiple sub-masks.

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.

The TOE supports the single Compliant power saving state of power off. Section 6.2.8 of [ST] states the TOE requires the validation of the authorization factor prior to allowing access to TSF data after exiting a Compliant power saving state.

2.1.17.2 Guidance Activities

(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.

Section "Setup and Configuration", subsection "TCG Opal Setting Try Limits" of [CCCG] describes how the administrator can set limits on the number of validation attempts allowed for PINs that have stable try limits.

(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.

Section "CC Operational Guidance", subsection "Validation - Try Limits and Persistence Settings" of [CCCG] identifies which validation failure limits an administrator may configure. The section includes a table identifying the default value of each validation failure limit.

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

The table in section "CC Operational Guidance", subsection "Validation - Try Limits and Persistence Settings" of [CCCG] identifies the authorization factors that allow the TOE to exit the Compliant power saving state (D3, or device full off).

2.1.17.3 KMD Activities

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.

Section 4.2 of [KMD] ("Authorization Failure Handling") describes the method the TOE uses to limit the number of consecutive failed authorization attempts. The TOE maintains a separate failure count for each authorization factor (PIN) that keeps track of the number of failed authentication attempts. The counter is reset to zero after a successful authentication. Non-persistent failure counters are reset to zero on power cycle. The persistence settings are set in the factory and are not configurable.

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).

Section 4.2 of [KMD] describes how the TOE validates authorization factors. The TOE validates the entered PIN by first calling the PBKDF with the PIN and the associated plaintext salt value as inputs. The output of the PBKDF is the ephemeral plaintext Authentication Key associated with the PIN. The TOE then calls the AES key unwrap function with the Authentication Key to unwrap the Transport Encryption Key (TEK) associated with the PIN. If the unwrap function produces the correct integrity check value (ICV), the PIN is valid, and authentication is successful. Otherwise, the PIN is invalid, and authentication is unsuccessful.

2.1.17.4 Test Activities

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.

The evaluator made six consecutive attempts to authorize to the TOE with an incorrect password. The login attempts were rejected five times and on the sixth attempt a lockout message was sent. The evaluator then attempted to login with a correct password and verified that a lockout message was received again.

Test 2: 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.

After being locked out the evaluator power cycled the TOE. After power cycling another login attempt was made. This time the TOE accepted the login attempt with a correct password.

2.2 User Data Protection (FDP)

2.2.1 Protection of Data on Disk (FDP_DSK_EXT.1)

2.2.1.1 TSS Activities

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.

Section 6.1 of [ST] ("Overview of TOE Operations") states the TOE accepts NVMe commands to read or write user data in its user-addressable non-volatile memory space. The TOE uses one in-line encryption engine for each port, employing AES in XTS mode with a 256-bit encryption key to encrypt all data prior to being written to user-addressable non-volatile memory and to decrypt all data as it is read from the same memory. The encryption engines are always in operation and cannot be disabled.

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 TOE does not rely on or invoke any cryptographic functions provided by the Operational Environment.

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.

Section 6.1 of [ST] states the TOE accepts NVMe commands to read or write user data in its user-addressable non-volatile memory space. The TOE uses one in-line encryption engine for each port, employing AES in XTS mode with a 256-bit encryption key to encrypt all data prior to being written to user-addressable non-volatile memory and to decrypt all data as it is read from the same memory. The encryption engines are always in operation and cannot be disabled.

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.

Section 6.4 of [ST] ("User Data Protection") states the TOE encrypts user data by default and without user intervention. Section 6.4.1 of [ST] ("Protection of Data on Disk (FDP_DSK_EXT.1)") states the TOE does not restrict reading or writing data to the drive until a user takes ownership using a TCG controller. The action of taking ownership locks the drive and constitutes the initialization process providing data-at-rest protection.

Section 6.4.1 of [ST] states the TOE supports three categories of storage: unencrypted for OS use; unencrypted for drive use (termed the system area); and encrypted. The unencrypted for OS use includes a shadow Master Boot Record used for boot. The OS is unable to access the system area.

2.2.1.2 Guidance Activities

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.

Section “Setup and Configuration” of [CCCG] describes the initial steps needed to enable the FDE function. Section "CC Operational Guidance”, subsection “Protection of Data on Disk & Specification of Management Functions” of [CCCG] states all Seagate Self Encrypting Drives (SEDs) are manufactured and delivered with encryption enabled and with the drive unlocked. Once the administrator takes ownership of the drive (the process described in the “Setup and Configuration” section of [CCCG]), all PINs controlling access to the encrypted user data areas on the drive are set to administrator-configured values.

2.2.1.3 KMD Activities

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.

Section 3 of [KMD] (“Encryption Engine”) describes the TOE’s data encryption engine, its components, and its implementation. The physical boundary of the TOE is the physical drive enclosure. The physical interface to the TOE is the PCIe connector and jumper block pins. The logical interface is the industry standard NVMe and Opal SSC protocols, carried on the PCIe interface. The TOE includes a Phison PS5020-E20 ASIC SSD Controller, which includes a built-in hardware AES encryption engine and controls all memory components directly (i.e., NAND flash and DRAM). The PS5020-E20 provides cryptographic services using NIST-validated algorithms. Figure 3 of [KMD] (“Encryption Engine Hardware Components: SSD ASIC PS5020-E20”) provides a block diagram of the functional components of the encryption engine, while Figure 2 of [KMD] (“Key Formation Diagram”) shows the data paths between functional components. It shows the data encryption engine controls all memory and there are no paths to non-volatile memory that do not pass through the 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).

Section 3.3 of [KMD] (“Operation”) states the TOE encrypt all user data by default. The TOE restricts user data reads and writes once a user takes ownership using a TCG controller. Section 3.3 of [KMD] states the TOE supports three categories of storage: unencrypted for OS use; unencrypted for drive use (termed the system area); and encrypted. The unencrypted for OS use includes a shadow Master Boot Record used for boot. The OS is unable to access the system area. When the TOE receives data from the host platform via PCIe connectors the encryption engine encrypts the data before it is written into non-volatile memory.

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.

Section 1 of [KMD] (“Introduction”) states the TOE encrypts user data by default in the out-of-the-box configuration. Access to data is not restricted until a user takes ownership via a TCG controller. After a user takes ownership, an authentication key is needed to unlock the drive, to read and write encrypted data. Section 3.2 of [KMD] (“Initialization and Configuration”) describes the startup process for the TOE. Section 2.5 of [KMD] (“Key Hierarchy”) describes the process for taking ownership of the drive.

The TOE developer provides tools to allow for inspection of the encrypted drive.

2.2.1.4 Test Activities

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.

The evaluator verified that the TOE was initialized and that encryption was enabled for the TOE’s Global address range. The TOE’s DRBG functionality was invoked to generate a random character pattern.

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.

The random data pattern above was written to the TOE.

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.

As demonstrated in FCS_CKM.4(a) testing the TOE's encryption keys were cleared. A search for the data pattern written above showed that it could not be found on the TOE.

2.3 Security Management (FMT)

2.3.1 Specification of Management Functions (FMT_SMF.1)

2.3.1.1 TSS Activities

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

Section 6.3.1 of [ST] ("Specification of Management Functions (FMT_SMF.1)") states the TOE changes a DEK when re-provisioning or when commanded. The TOE generates each DEK using its implementation of HMAC_DRBG.

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

Section 6.2.2 of [ST] ("Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), and FCS_CKM_EXT.6)") describes how the TOE cryptographically erases the DEK from volatile memory (i.e., as specified in FCS_CKM.4(a)). The TOE destroys DEKs in volatile memory when power is removed, when the drive is locked, and when the TOE generates a new DEK to erase a locking range. When the TOE is powered off, all keys are destroyed. When the drive is locked, the TOE overwrites all keys with zeros. When the TOE generates a new DEK to erase a locking range, the TOE overwrites the existing DEK with the new value of 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.

Section 6.3.1 of [ST] states the TOE initiates a firmware update using the Firmware Image Download and Firmware Image Commit commands.

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.

Section 6.3.1 of [ST] describes the management function to configure a password for firmware update. The SID is the password required for firmware updates. The initial value of the SID is the 32-byte

Manufacturer's SID (MSID), a public drive-unique value that is used as the default PIN. The drive must be "personalized" to change the initial value of the SID to private values. Once the administrator takes ownership of the drive, the SID value is set to the administrator-configured value. The commands to configure the SID value are NVME SECURITY SET PASSWORD, and TCG Set Method.

Section 6.3.1 of [ST] describes the management function to configure the number of failed validation attempts required to trigger corrective behavior (i.e., Try Limit). The Try Limit has a default value of either 5 or 100, depending on the version of firmware installed in the TOE.

2.3.1.2 Guidance Activities

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.

Section "Setup and Configuration", subsection "TCG Opal Gen Key" of [CCCG] provides instructions for changing the DEK using the Gen Key command.

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.

Section "CC Operational Guidance", subsection "Firmware Access Control and Firmware Trusted Update" of [CCCG] describes how to initiate TOE firmware 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.

Section "Setup and Configuration", subsection "TCG Opal Setup & Configuration" of [CCCG] describes the method by which the administrator sets new authorization factors when taking ownership of the TOE. Section 6.3.1 of [ST] ("Specification of Management Functions (FMT_SMF.1)") describes the TOE's default authorization factor, which consists of the Manufacturer's SID (MSID). The MSID is unique to each TOE instance and provides the default value for the User's Security Identifier (SID). The administrator sets a new value of the SID when taking ownership of the TOE.

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

The TOE does not provide a Key Recovery capability.

2.3.1.3 KMD Activities

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.

Section 4.1 of [KMD] ("DEK Import") states the TOE does not allow importing any DEK from outside the TOE.

2.3.1.4 Test Activities

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).

A vendor test tool was used to execute a Vendor Unique Command (VUC) to retrieve the TOE's DEK. The TCG revert command was executed to clear CSPs. After this the VUC was used again to retrieve the DEK which was shown to be different.

Additionally, the TOE's ability to clear the DEK (both the unencrypted one in volatile memory and the encrypted one in non-volatile memory) was demonstrated in greater detail in FCS_CKM.4(a).

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.

The evaluator used a vendor test tool to initiate a firmware update.

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.

The ability to configure failed validation attempt limits was demonstrated in the tests for FCS_VAL.

2.4 Protection of the TSF (FPT)

2.4.1 Firmware Access Control (FPT_FAC_EXT.1)

2.4.1.1 TSS Activities

The evaluator shall examine the TSS to ensure that it describes information stating how the Access Control process takes place along with a description of the values that are used.

Section 6.5.1 of [ST] ("Firmware Access Control and Update Authentication (FPT_FAC_EXT.1, FPT_FUA_EXT.1)") states the TOE's Firmware Access Control function requires the administrator to unlock the firmware download port in order to enable the firmware update to proceed. The administrator provides the Security Identifier (SID) credential to the TOE in order to unlock the firmware download port, using the following procedure:

1. Administrator opens a session to the Admin SP (Security Provider)
2. Administrator authenticates with Admin SP SID credential
3. Administrator unlocks the firmware download port
4. Administrator closes the session to the Admin SP.

The Admin SP SID credential is a password with a minimum length of 8 bytes and a maximum length of 32 bytes. The administrator sets the Admin SP SID after initializing the TOE.

2.4.1.2 Guidance Activities

The evaluator ensures that the Operational Guidance describes how the user will be expected to interact with the authorization process.

Section "CC Operational Guidance", subsection "Firmware Access Control and Firmware Trusted Update" of [CCCG] describes how the user interacts with the authorization process.

2.4.1.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.1.4 Test Activities

The evaluator shall perform the following test.

Test 1: The evaluator shall try installing a firmware upgrade and verify that a prompt is required and the appropriate value is necessary for the update to continue.

A vendor test tool was used to verify that the TOE will not accept a firmware update through a locked firmware port and will accept an update through an unlocked port.

2.4.2 Firmware Update Authentication (FPT_FUA_EXT.1)

2.4.2.1 TSS Activities

The evaluator shall examine the TSS to ensure that it describes how the TOE uses the RTU, what type of key or hash value, and where the value is stored on the RTU. The evaluator shall also verify that the TSS contains a description (storage location) of where the original firmware exists.

Section 6.5.1 of [ST] ("Firmware Access Control and Update Authentication (FPT_FAC_EXT.1, FPT_FUA_EXT.1)") states the TOE authenticates the source of a firmware update using the PKCS #1, v2.1 RSA digital signature algorithm, with a key size (modulus) of 4096 bits. The mechanism uses the Root of Trust for Update (RTU) key stored in ROM that contains the hashed public key. This key is used to verify the public key included with the signed firmware binary file. After successful verification, the public key in the binary will then be used to verify the signature on an update image. If the verification fails an error is returned and the update is not performed. The firmware key store and the signature verification algorithm are stored in a write protected area on the TOE.

2.4.2.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.4.2.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.2.4 Test Activities

There are no test evaluation activities for this SFR.

2.4.3 Protection of Key and Key Material (FPT_KYP_EXT.1)

2.4.3.1 TSS Activities

Modified in accordance with TD0458.

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

Section 6.5.2 of [ST] (“Protection of Key and Key Material (FPT_KYP_EXT.1)”) states the TOE stores keys in non-volatile memory only when the keys have been wrapped. The TOE performs key wrapping using AES in KW mode.

2.4.3.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.4.3.3 KMD Activities

Modified in accordance with TD0458.

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.

Section 2 of [KMD] (“Keys and Key Hierarchy”) describes the keys used by the TOE in the key hierarchy and their storage in volatile and non-volatile memory. Section 2.3 of [KMD] (“Intermediate Keys”) describes the storage location within the TOE of all keys in the key hierarchy from the BEV down to the KEK. Of these keys, the TOE stores only the TEK and KEK in non-volatile memory, both of which are protected using AES key wrapping. Section 2.4 of [KMD] (“Data Encryption Key”) describes the DEK and states the TOE encrypts the DEK when storing it in non-volatile memory. More specifically, Section 2.5 of [KMD] (“Key Hierarchy”) states the TOE protects the DEK using AES key wrapping when storing it in non-volatile memory.

The evaluator reviewed the description of the key chain in Section 2 of [KMD] and confirmed the TOE employs the method selected in FPT_KYP_EXT.1, of using key wrapping as specified in FCS_COP.1(d), when storing keys in non-volatile memory. The TOE does not store plaintext keys in non-volatile memory.

2.4.3.4 Test Activities

There are no test evaluation activities for this SFR.

2.4.4 Power Saving States (FPT_PWR_EXT.1)

2.4.4.1 TSS Activities

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

Section 6.5.3 of [ST] (“Power Saving States and Timing (FPT_PWR_EXT.1, FPT_PWR_EXT.2)”) states the TOE supports a single Compliant power state of device full off (D3).

2.4.4.2 Guidance Activities

Modified in accordance with TD0460.

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.

Section “CC Operational Guidance”, subsection “Power Saving States and Timing of Power Saving States” of [CCCG] states the TOE supports a single Compliant power state of device full off (D3). The TOE has two

possible power transitions: power off to on; and power on to off. The TOE does not support any other power saving states.

2.4.4.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.4.4 Test Activities

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.

Covered under testing by FCS_CKM.4(b). The TOE destroys keys conformant to the selection in FCS_CKM_EXT.6 in the ST. In FCS_CKM.4(b), the evaluator shows that the keys are destroyed and verifies this by doing a search of the key in parts and as a whole on the TOE. The keys are not found after destruction and the test is a pass.

2.4.5 Timing of Power Saving States (FPT_PWR_EXT.2)

2.4.5.1 TSS Activities

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

Section 6.5.3 of [ST] (“Power Saving States and Timing (FPT_PWR_EXT.1, FPT_PWR_EXT.2)”) states the TOE has two possible transitions: power off to on; and power on to off. The TOE transitions from power on to power off when the system removes power to the drive.

2.4.5.2 Guidance Activities

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).

Section “CC Operational Guidance”, subsection ”Power Saving States and Timing of Power Saving States” of [CCCG] states the TOE has two possible power transitions: power off to on; and power on to off. The TOE transitions from power on to power off when the system removes power to the drive. After power is removed, it takes approximately 2 seconds for DRAM volatile memory and about 30 mS for DRAM volatile memory to completely power down.

2.4.5.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.5.4 Test Activities

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.

The claimed Compliant power saving state for the TOE is D3. The only way to enter this state is removal of power from the TOE. FCS_CKM_EXT.6 specifies FCS_CKM.4(b) as the method of key destruction, which includes “removal of power from the TOE” in its claims in [ST]. Test 1 of FCS_CKM.4(b), which relates to destruction of keys stored in volatile memory, states that it is not performed in the case of

removal of power from the TOE because the destruction of keys in such a situation is an inherent part of the design of volatile memory in general.

2.4.6 Rollback Protection (FPT_RBP_EXT.1)

2.4.6.1 TSS Activities

The evaluator shall examine the TSS to ensure that it describes at a high level the process for verifying that security version checking is performed before an upgrade is installed. The evaluator shall verify that a high level description of the types of error codes are provided and when an error would be triggered.

Section 6.5.4 of [ST] ("Rollback Protection (FPT_RBP_EXT.1)") states the TOE supports the capability to ensure it is not possible to downgrade the TOE to a lower security version number. If a firmware update package is downloaded to the drive with an invalid firmware revision number, the TOE generates and returns an error code and rejects the firmware update package. The TOE generates the following error codes:

- Invalid Firmware Image—generated if the TOE determines the firmware image is invalid (i.e., the TOE cannot verify the digital signature on the firmware image)
- Trying to download older firmware over newer firmware—generated when the TOE determines the downloaded firmware image has an invalid revision number.

2.4.6.2 Guidance Activities

The evaluator ensures that a description is provided on how the user should interpret the error codes.

Section "CC Operational Guidance", subsection "Firmware Rollback Protection" of [CCCG] describes the process by which the TOE ensures it is not possible to downgrade the TOE to a lower security version number.

2.4.6.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.6.4 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall try installing a lower security version number upgrade (either by just modifying the version number or by using an upgrade provided by the vendor) and will verify that the lower version cannot be installed and an error is presented to the user.

The evaluator attempted to install a lower versioned number of the TOE firmware. This update attempt was rejected by the TOE.

2.4.7 TSF Testing (FPT_TST_EXT.1)

2.4.7.1 TSS Activities

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

Section 6.5.5 of [ST] ("TSF Testing (FPT_TST_EXT.1)") lists the known answer tests for cryptographic functions and describes how the TOE performs the known answer tests. For each cryptographic known

answer test, the TOE uses known inputs to calculate an expected cryptographic result and compares that result to the expected or known result for the known input. If the calculated result matches the expected result, the test passes; if it does not match, the test fails.

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.

Section 6.5.5 of [ST] states the TOE performs a Firmware Integrity Check as part of the secure boot process. The TOE first loads the firmware from non-volatile memory into volatile memory using routines in Read Only Memory (ROM). The TOE then verifies the RSA signature of the firmware in volatile memory using firmware routines and the public key in ROM. If the signature is verified to be correct, the ROM code transfers control to the firmware in volatile memory. If the signature does not verify, then the TOE indicates a fatal error.

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.

Section 6.5.5 of [ST] states the TOE performs health tests for al deterministic random bit generation services consistent with section 11.3 of NIST SP 800-90A. The TOE compares each newly generated random number to the previously generated number. The test fails if they are equal. The TOE also compares newly generated entropy data with the previously generated entropy data. Again, the test fails if they are equal.

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

Section 6.5.5 of ST describes the following known answer tests (KATs) for the cryptographic functions the TOE implements:

- RSA signature verification KAT (FCS_COP.1(a))
- SHA-256 and SHA-512 cryptographic hash KAT (FCS_COP.1(b))
- HMAC-SHA-256 message authentication KAT (FCS_COP.1(c))
- AES Key Wrap and Unwrap KATs (FCS_COP.1(d))
- AES-XTS-256 data encryption and decryption KATs (FCS_COP.1(f)).

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 start-up.

Section 6.5.5 of [ST] states the TSF performs RSA signature verification of a new firmware image before the image can be loaded. The TSF accepts the new firmware only if the signature is verified. This test is run whenever the TOE downloads a new firmware image.

2.4.7.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.4.7.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.7.4 Test Activities

There are no test evaluation activities for this SFR.

2.4.8 Trusted Update (FPT_TUD_EXT.1)

2.4.8.1 TSS Activities

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.

Section 6.5.6 of [ST] ("Trusted Update (FPT_TUD_EXT.1)") states the TOE verifies TOE updates prior to installation using the RSA digital signature algorithm with a key size (modulus) of 4096 bits. The updates are digitally signed by Seagate.

Section 6.5.1 of [ST] ("Firmware Access Control and Update Authentication (FPT_FAC_EXT.1, FPT_FUA_EXT.1)") states the firmware key store and the signature verification algorithm are stored in a write protected area on the TOE. The firmware can only be updated using the authenticated update mechanism by an authorized user where the authorized source that signs TOE updates is Seagate. The TOE authenticates the source of the firmware update using the RSA digital signature algorithm, with a key size (modulus) of 4096 bits. The mechanism uses the Root of Trust for Update (RTU) key stored in ROM that contains the public key to verify the signature on an update image. An error code is returned if any part of the firmware update process fails. The TOE only allows installation of an update if the digital signature has been successfully verified.

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.

All cryptographic operations, including signature verification, are performed by the TOE.

2.4.8.2 Guidance Activities

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.

Section "CC Operational Guidance", subsection "Firmware Access Control and Firmware Trusted Update" of [CCCG] describes: how the TOE obtains vendor updates to the TOE, by obtaining a genuine Seagate Secure firmware update package from the Seagate support web site; how the TOE verifies the digital signature on the firmware update package using RSA as defined in FCS_COP.1(a); and the actions that take place for successful and unsuccessful signature verification.

2.4.8.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.4.8.4 Test Activities

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.

The evaluator verified the TOE's version before and after and update and confirmed that the new version matched the update that was installed.

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.

After installing a legitimate update the evaluator verified that the TOE could be initialized and write encrypted data.

3 Security Assurance Requirements

3.1 Class ASE: Security Target Evaluation

An evaluation activity is defined here for evaluation of Exact Conformance claims against a cPP in a Security Target. Other aspects of ASE remain as defined in the CEM.

3.1.1 Conformance Claims (ASE_CCL.1)

The table below indicates the actions to be taken for particular ASE_CCL.1 elements in order to determine exact conformance with a cPP.

ASE_CCL.1 element	Evaluator Action
ASE_CCL.1.8C	The evaluator shall check that the statements of security problem definition in the PP and ST are identical.
ASE_CCL.1.9C	The evaluator shall check that the statements of security objectives in the PP and ST are identical.
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.

3.1.1.1 ASE_CCL.1.8C

Section 3 of [ST] includes by reference the security problem definition from [CPP_FDE_EE_V2.0E]. As such, the security problem definition in [ST] is identical to the statement of security problem definition specified in [CPP_FDE_EE_V2.0E].

3.1.1.2 ASE_CCL.1.9C

Section 4 of [ST] includes by reference the statement of security objectives from [CPP_FDE_EE_V2.0E]. As such, the statement of security objectives in [ST] is identical to the statement of security objectives specified in [CPP_FDE_EE_V2.0E].

3.1.1.3 ASE_CCL.1.10C

Section 5 of [ST] states the security functional requirements included in the ST are drawn from [CPP_FDE_EE_V2.0E].

The evaluator examined the requirements specified in Section 6 of [ST] and confirmed the following:

- All mandatory functional requirements from [CPP_FDE_EE_V2.0E] are included in [ST] through reproduction.

- All selection-based requirements from [CPP_FDE_EE_V2.0E] indicated by selections made in the mandatory requirements are included in [ST] through reproduction
- No requirements not specified in [CPP_FDE_EE_V2.0E] have been specified in [ST].

3.2 Class ADV: Development

3.2.1 Basic Functional Specification (ADV_FSP.1)

The EAs for this assurance component focus on understanding the interfaces (e.g., application programing interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Sections 2, 3, and 4, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

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

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

CEM ADV_FSP.1 Work Units	Evaluation Activities
ADV_FSP.1-1 The evaluator shall examine the functional specification to determine that it states the purpose of each SFR-supporting and SFR-enforcing TSFI.	Evaluation Activity: The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.
ADV_FSP.1-2 The evaluator shall examine the functional specification to determine that the method of use for each SFR-supporting and SFR-enforcing TSFI is given.	Evaluation Activity: The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.
ADV_FSP.1-3 The evaluator shall examine the presentation of the TSFI to determine that it identifies all parameters associated with each SFR-enforcing and SFR supporting TSFI.	Evaluation Activity: The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

CEM ADV_FSP.1 Work Units	Evaluation Activities
ADV_FSP.1-4 The evaluator shall examine the rationale provided by the developer for the implicit categorisation of interfaces as SFR-non-interfering to determine that it is accurate.	Paragraph 561 from the CEM: "In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, this work unit should be considered satisfied." Since the rest of the ADV_FSP.1 work units will have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.
ADV_FSP.1-5 The evaluator shall check that the tracing links the SFRs to the corresponding TSFIs.	5.2.1.3 Evaluation Activity: The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.
ADV_FSP.1-6 The evaluator shall examine the functional specification to determine that it is a complete instantiation of the SFRs.	EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are covered. Therefore, the intent of this work unit is covered
ADV_FSP.1-7 The evaluator shall examine the functional specification to determine that it is an accurate instantiation of the SFRs.	EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e., at the TSFI) are addressed, and that the description of the interfaces is accurate with respect to the specification captured in the SFRs. Therefore, the intent of this work unit is covered.

3.2.1.1 Evaluation Activity

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

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

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

The TOE is a set of Seagate self-encrypting drives. As described in **Error! Reference source not found.**, an end user interacts with the TOE through a TCG host controller. Seagate SEDs implement TCG Enterprise

Security Subsystem Class (SSC), TCG Opal SSC, and ATA Security specifications. These specifications determine TOE behavior including the interfaces each Seagate SED presents to host controllers.

Error! Reference source not found. identifies TCG methods and ATA Security commands relevant to the TSF. In particular, **Error! Reference source not found.** section "Setup and Configuration" describes the steps necessary to put a Seagate SED into the evaluated configuration. Each step includes examples to illustrate interactions between a host controller and a SED. The examples are based on Seagate TCG and ATA Security libraries. The code samples would be modified to work with any customer specific TCG Opal implementations.

The evaluator used **Error! Reference source not found.** and **Error! Reference source not found.**] to identify TCG methods and ATA Security commands relevant to each TOE security function. **Error! Reference source not found.** section 6.2.2 "Cryptographic Key Destruction (FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), FCS_CKM_EXT.6)" provides convenient labels for most TOE security functions. Examples in **Error! Reference source not found.** section "Setup & Configuration" show additional configuration functions. The tables in the [CCCG] sections "TCG Opal Security Mode Services" and "Non-Security Mode Services" contain security service names along with TCG methods and ATA Security comments for TOE security functions.

The table below contains a mapping of SFRs to Security Functions.

Mapping from SFR to Security Functions

SFR	Security Function
FCS_CKM.1(b)	No external interface
FCS_CKM.1(c)	No external interface
FCS_CKM.4(a)	Device full off
FCS_CKM.4(b)	No external interface
FCS_CKM_EXT.4(a)	No external interface
FCS_CKM_EXT.4(b)	Device full off
FCS_CKM_EXT.6	Set PIN
FCS_CKM_EXT.6	Reset Module
FCS_CKM_EXT.6	Exit CC Security Mode
FCS_CKM_EXT.6	Enable / Disable Admin SP Admin(s)
FCS_CKM_EXT.6	Enable / Disable Locking SP Admin(s), non- SUDR User(s)
FCS_CKM_EXT.6	Lock / Unlock User Data Range for Read and/or Write
FCS_COP.1(a)	No external interface
FCS_COP.1(b)	No external interface
FCS_COP.1(c)	No external interface
FCS_COP.1(d)	No external interface
FCS_COP.1(f)	No external interface
FCS_KDF_EXT.1	No external interface
FCS_KYC_EXT.2	No external interface

SFR	Security Function
FCS_RBG_EXT.1	No external interface
FCS_SNI_EXT.1	No external interface
FCS_VAL_EXT.1	No external interface
FDP_DSK_EXT.1	User Data Read / Write
FMT_SMF.1	Cryptographic Erase of non- SUDR
FMT_SMF.1	Cryptographic Erase of SUDR
FMT_SMF.1	Firmware Download
FPT_FAC_EXT.1	Firmware Download
FPT_FUA_EXT.1	Firmware Download
FPT_KYP_EXT.1	No external interface
FPT_PWR_EXT.1	No external interface
FPT_PWR_EXT.2	Device full off
FPT_RBP_EXT.1	Firmware Download
FPT_TST_EXT.1	Reset Module
FPT_TUD_EXT.1	Firmware Download

3.2.1.2 Evaluation Activity

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

Please see section **Error! Reference source not found.**, which summarizes the evaluation team's examination of TOE security function interfaces. The evaluation team used the mappings to confirm **Error! Reference source not found.** and the public specifications adequately identify and describe the parameters for each TOE security function interface.

3.2.1.3 Evaluation Activity

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

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2 (*Evaluation Activities for SFRs*), as well as any applicable EAs in Sections 3 and 4, including the EAs associated with testing of the interfaces.

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

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

Please see the table in section 3.2.1.1 above.

3.3 Class AGD: Guidance Documents

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

3.3.1 Operational User Guidance (AGD_OPE.1)

Specific requirements and checks on the user guidance documentation are identified (where relevant) in the individual Evaluation Activities for each SFR, and for some other SARs (e.g. ALC_CMC.1).

3.3.1.1 Evaluation Activity

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).

Error! Reference source not found. section "Setup and Configuration" describes the steps necessary to put the TOE into the evaluated configuration.

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.

The NIAP Product Compliant List entry for this evaluation includes a copy of [CCCG].

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 TOE does not claim support for multiple operational environments.

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.

Each Seagate SED firmware and hardware contain the drive's cryptographic engine, which is not configurable.

The TOE is a set of Seagate SEDs, which rely on the host platform for power management so configuring the operational environment for this is not applicable.

Error! Reference source not found. sections "Power Saving States and Timing of Power Saving States" and "Cryptographic Key and Key Material Destruction (Power Management)" provide sufficient information for a host platform to manage a Seagate SED's power states.

The TCG Storage and ATA Security Specifications identified in section **Error! Reference source not found.** specify interfaces and behaviors for self-encrypting drives, which include the Seagate SEDs. **Error! Reference source not found.** serves to identify the interfaces and behaviors related to the security functional requirements and security functions in **Error! Reference source not found.**.

3.3.2 Preparative Procedures (AGD_PRE.1)

As for the operational guidance, specific requirements and checks on the preparative procedures are identified (where relevant) in the individual Evaluation Activities for each SFR.

3.3.2.1 Evaluation Activity

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.

Error! Reference source not found. section "Introduction" states the need for host system that supports TCG security mode commands. Section "Setup and Configuration" covers TCG Opal devices.

Error! Reference source not found. section "Protection of Data on Disk & Specification of Management Functions" covers managing security of Seagate SEDs. The section addresses personalization of a drive, interaction with an Authorization Acquisition component, supporting multiple users, and erasing a drive.

Error! Reference source not found. section "TCG Opal Security Mode Services" identifies security services along with access restrictions on those services.

3.4 Class ALC: Life-Cycle Support

3.4.1 Labeling of the TOE Assurance Activity (ALC_CMC.1)

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

The evaluator examined the TOE over video during remote testing and observed it is labelled with a unique reference in the form of a serial number and identifiers to match the model number and firmware in the Security Target.

3.4.2 TOE CM Coverage Assurance Activity (ALC_CMS.1)

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

The evaluator confirmed via the Security Target that the developer provided an identification of the TOE defined as: Product Name, model number, standard, and firmware. **Error! Reference source not found.** uniquely identifies each of **Error! Reference source not found.**, **Error! Reference source not found.**, and **Error! Reference source not found.** by title, version and date.

3.5 Class ATE: Tests

3.5.1 Independent Testing – Conformance (ATE_IND.1)

Testing is performed to confirm the functionality described in the TSS as well as the operational guidance documentation. The focus of the testing is to confirm that the requirements specified in the SFRs are being met.

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

The SFR-related Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The tests identified in these other Evaluation Activities constitute a sufficient set of tests for the purposes of meeting ATE_IND.1.2E. It is important to note that while the Evaluation Activities identify the testing that is necessary to be performed, the evaluator is responsible for ensuring that the interfaces are adequately tested for the security functionality specified for each SFR.

3.5.1.1 Evaluation Activity

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 examined the TOE over video during remote testing and confirmed that it conforms with the hardware, configuration and firmware described in the ST.

3.5.1.2 Evaluation Activity

The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state

The test tool used the evaluate the TOE performed Device Under Test verification activities prior to testing. The evaluator confirmed it presented no errors and entered a running state. The evaluator performed a version and model verification activity prior to testing.

3.5.1.3 Evaluation Activity

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.

The evaluator prepared a test plan prior to testing outlining the required test activities to be performed. Throughout testing the test plan was updated with results to eventually become the test report.

The test plan laid out a subset of all instances of the TOE in the evaluation to be tested. Similar instances of the TOE were grouped together based on hardware. Each instance of the TOE did not have any variations with software dependencies, software binaries, libraries used, management interfaces or functional differences so hardware only needed to be considered. When an instance or instances of the TOE were not tested justification was provided in the test report via equivalency rationale.

The test report lists each instance of the TOE tested for each SFR and details it as defined in the ST.

The test plan established and set of procedures to follow with steps and configuration necessary to achieve the expected result.

The test report describes in detail the activities the evaluator performed along with actual results in the form of evidence to accomplish each test. Each test account is accompanied by a test result in the form of 'pass' or 'fail'. The test report also establishes an overall result for the cumulative test activities stated by a 'pass' or 'fail'.

3.6 Class AVA: Vulnerability Assessment

3.6.1 Vulnerability Survey (AVA_VAN.1)

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

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

CEM AVA_VAN.1 Work Units	Evaluation Activities
AVA_VAN.1-1 The evaluator shall examine the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.	The evaluator shall perform the CEM activity as specified. *If the iTC specifies any tools to be used in performing this analysis in section A.3.4, the following text is also included in this cell: "The calibration of test resources specified in paragraph 1418 of the CEM applies to the tools listed in Appendix A, Section A.1.4."*
AVA_VAN.1-2 The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state.	The evaluator shall perform the CEM activity as specified.
AVA_VAN.1-3 The evaluator shall examine sources of information publicly available to identify potential vulnerabilities in the TOE.	Replace CEM work unit with activities outlined in Appendix A, Section A.1.
AVA_VAN.1-4 The evaluator shall record in the ETR the identified potential vulnerabilities that are candidates for testing and applicable to the TOE in its operational environment.	Replace the CEM work unit with the analysis activities on the list of potential vulnerabilities in Appendix A, section A.1, and documentation as specified in Appendix A, Section A.3.
AVA_VAN.1-5 The evaluator shall devise penetration tests, based on the independent search for potential vulnerabilities.	Replace the CEM work unit with the activities specified in Appendix A, section A.2.
AVA_VAN.1-6 The evaluator shall produce penetration test documentation for the tests based on the list of potential vulnerabilities in sufficient detail to enable the tests to be repeatable. The test documentation shall include: a) identification of the potential vulnerability the TOE is being tested for; b) instructions to connect and setup all required test equipment as required to conduct the penetration test; c) instructions to establish all penetration test prerequisite initial conditions; d) instructions to stimulate the TSF; e) instructions for observing the behaviour of the TSF;	The CEM work unit is captured in Appendix A, Section A.3; there are no substantive differences.

CEM AVA_VAN.1 Work Units	Evaluation Activities
f) descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results; g) instructions to conclude the test and establish the necessary post-test state for the TOE.	
AVA_VAN.1-7 The evaluator shall conduct penetration testing.	The evaluator shall perform the CEM activity as specified. See Appendix A, Section A.3 for guidance related to attack potential for confirmed flaws.
AVA_VAN.1-8 The evaluator shall record the actual results of the penetration tests.	The evaluator shall perform the CEM activity as specified.
AVA_VAN.1-9 The evaluator shall report in the ETR the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results.	Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.
AVA_VAN.1-10 The evaluator shall examine the results of all penetration testing to determine that the TOE, in its operational environment, is resistant to an attacker possessing a Basic attack potential.	This work unit is not applicable for Type 1 and Type 2 flaws (as defined in Appendix A, Section A.1), as inclusion in this Supporting Document by the iTC makes any confirmed vulnerabilities stemming from these flaws subject to an attacker possessing a Basic attack potential. This work unit is replaced for Type 3 and Type 4 flaws by the activities defined in Appendix A, Section A.3.
AVA_VAN.1-11 The evaluator shall report in the ETR all exploitable vulnerabilities and residual vulnerabilities, detailing for each: a) its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication); b) the SFR(s) not met; c) a description; d) whether it is exploitable in its operational environment or not (i.e. exploitable or residual). e) the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities, and the corresponding values using the tables 3 and 4 of Annex B.4.	Replace the CEM work unit with the reporting called for in Appendix A, Section A.3.

Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in Appendix A, while an “outline” of the assurance activity is provided below.

3.6.1.1 AVA_VAN.1 Evaluation Activity

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.

Section 2.2 of [ST] identifies the TOE in terms of the device type (SSC Opal), firmware (PS5020-E20), processor (ARMv7-R ARM Cortex-R5), and specific product names with associated model numbers and firmware identifiers.

3.6.1.2 AVA_VAN.1 Evaluation Activity

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 formulated the flaw hypotheses used as a basis for the vulnerability search as described in section A.1 of [SD]. Specifically, the following search terms were considered for Type 1 flaw hypotheses based on information provided in [ST]:

- Vendor/developer/product names:
  - “Seagate”
  - “Phison”
  - “Nytro”
- Underlying components as required by [SD] for EE SED products:
  - “PS5020-E20”
  - “Arm Cortex-R5”
  - “ARMv7-R”
  - “self encrypting drive”
  - “opal”
- Search terms specified in [SD] for general FDE technology and for EE products specifically:
  - “drive encryption”
  - “disk encryption”
  - “key destruction”
  - “key sanitization”

Per iTC guidance in [SD], the following vulnerability repositories were searched:

- Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/
- National Vulnerability Database: https://nvd.nist.gov/

- US-CERT: http://www.kb.cert.org/vuls/html/search

The only Type 2 hypothesis identified in [SD] relates to software FDE. Because the TOE is a hardware SED, this flaw hypothesis does not apply.

The evaluator also examined the vendor security advisory site at https://www.seagate.com/support/security/ to identify any potential security flaws in Seagate branded products that were not present in the repositories linked above. The evaluator performed several iterations of vulnerability searches, most recently on April 4, 2024. Through this process it was determined that no residual publicly-disclosed vulnerabilities are present in the TOE.