Assurance Activity Report

Novachips Co., Ltd
Scalar and Express P-series SSD,
version NV.R1900

VID 11262

UL13480549-AAR Rev1.3
June 7, 2022

Evaluated by:

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

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

TOE Evaluation Sponsor and Developer

Novachips Co., Ltd
5F, B tower, Global Convergence Center, 46 Dallaenae-ro,
Sujeong-gu, Seongnam-si, Gyeonggi-do, 13449, South Korea

ST Author

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

Evaluation Personnel

Oleg Andrianov
Michael C. Baron

Applicable Common Criteria Version

CC Version 3.1 R5, April 2017

Common Evaluation Methodology Version

CEM Version 3.1 R5, April 2017

Applicable Common Criteria Protection Profiles

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

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

Table of Contents

1 Overview

This document presents evaluation results of the Scalar and Express P-series SSD, version NV.R1900 against the collaborative collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition Version 2.0 + Errata 20190201 February 1, 2019 [AA] and collaborative Protection Profile for Full Drive Encryption - Encryption Engine Version 2.0 + Errata 20190201 [EE]. This document contains a description of the assurance activities and associated results as performed by UL, an accredited Common Criteria Testing Laboratory. This Evaluation was conducted with the oversight and guidance provided by the National Information Assurance Partnership and its contributors.

1.1 Test Equivalency

The evaluator performed testing on TOE models NS371P10T0CC0-1F and NS561P500GCE7-1F. The [ST] claims 7 models.

Part number	HW Version	Host Interface	Firmware Source	PCB & BOM	User Capacity	ASIC Controller	NAND Flash
NS361P500GCCR-1F	04MB3	SATA / AHCI	NV.R1900_1000	2.5"_legacy_4n	500GB	NVS3800-39	1Tb NV-NAND x4
NS371P02T0CC1-1F	08MN3	SATA / AHCI	NV.R1900_1000	2.5"Hlsingle_8n	2TB	NVS3800-39	2Tb HL-NAND x8
NS371P04T0CC1-1F	16MN3	SATA / AHCI	NV.R1900_1000	2.5"_HLlsingle_16n	4TB	NVS3800-39	2Tb HL-NAND x16
NS371P10T0CC0-1F	16MN3	SATA / AHCI	NV.R1900_1000	2.5"_HLdual_40n	10TB	NVS3800-39	2Tb HL-NAND x40
NS561P500GCE7-1F	02MB3	PCIe / NVMe	NV.R1900_1000	M.2_legacy_4n	500GB	NVS3800-59	1Tb NV-NAND x4
NS571P02T0CK7-1F	16SN3	PCIe / NVMe	NV.R1900_1000	M.2_HL22110_8n	2TB	NVS3800-59	2Tb HL-NAND x8
NS571P08T0CC0-1F	16MN3	PCIe / NVMe	NV.R1900_1000	U.2_Hldual_32n	8TB	NVS3800-59	2Tb HL-NAND x32

All listed models are built upon a single SSD controller design architecture and are running the same firmware. Those models considered to be equivalent to tested models. Differences in interface and controller models were covered by tested models.

1.2 Test Environment

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

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

The test facility was prepared by the vendor and inspected by UL and physically located at Novachip's facility in an access-controlled room.

1.2.1 Test Equipment

- Real View ULINK2 Debug controller for JTAG connection
- PCIe Bus Analyzer LeCroy Summit T34
- PCIe Active Interposer Interface
- SATA Bus Analyzer LeCroy Sierra M6-2
- M.2 PCIe Adapter
- Teledyne LeCroy SAS SATA Protocol Suite Version: 6.10 build 1230
- Teledyne LeCroy PCIe Protocol Analysis Version: 8.78 build 3035
- Novachips Test Tool – CryptoOfficer GUI tool Version: 1.7
- Novachips Test Tool – CryptoOfficerCLI tool Version: 1.7
- uVision JTAG Version: V4.60.0.0

- HxD Hex Editor Version: 1.7.7.0
- Hex2bin 2.5
- HxD Hex Editor Version 2.5.0.0.

1.2.2 Test Scripts

- CKM.4 Key location tool (internal tool)

1.2.3 Test Firmware

- Novachips Firmware update tool NCMPTool Version: NV.R1900_1000
- Novachips Firmware update tool NCMPTool Version: NV.R1900_0999 (test firmware images)
- Novachips Firmware update tool NCMPTool NV.R1900_1001 (test firmware images)

1.3 Technical Decisions

This section lists the Technical Decisions issued by NIAP relevant to the Protection Profiles associated with the evaluation. A description as to the applicability to the TOE, is provided.

Table 1: Technical Decisions Collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition		
TD	TD Title	TOE Applicability
0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No
0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes

Table 2: Technical Decisions Collaborative Protection Profile for Full Drive Encryption – Encryption Engine		
TD	TD Title	TOE Applicability
0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No
0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Yes
0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Yes
0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes

1.4 Cryptographic Algorithms Validation

The TOE has cryptographic algorithms that have been validated according to the National Institute of Standards and Technology (NIST) Cryptographic Algorithm Validation Program (CAVP).

Operation	**SFR**	**Algorithm**	**NIST Standard**	**Certificate**
Cryptographic Key Generation	FCS_CKM.1(b)	DRBG	SP800-90A	C463
Cryptographic Key Generation	FCS_CKM.1(c)	DRBG	SP800-90A	C463
Signature Verification	FCS_COP.1(a)	ECDSA	FIPS PUB 186-4	A897

Hash Algorithm	FCS_COP.1(b)	SHS	FIPS 180-4	C411
Hash Algorithm	FCS_COP.1(b)	SHS	FIPS 180-4	A897
Keyed-Hash Message Authentication Code	FCS_COP.1(c)	HMAC	FIPS 198-1	A897
Block Cipher Encryption	FCS_COP.1(d)	AES-KW	SP 800-38F	A897
Block Cipher Encryption	FCS_COP.1(f)	AES-XTS	SP 800-38E	C448
Key Derivation	FCS_KDF_EXT.1	KDF	SP 800-132	A897
Cryptographic Key Generation	FCS_RBG_EXT.1	DRBG	SP800-90A	C463

2 SFR Evaluation Activities and Results

This section describes the evaluation activities associated with the SFRs included in [ST] and the results of those activities as performed by the CCTL evaluation team. The evaluation activities are defined in [AASD] and [EESD]. NIAP Technical Decisions have been applied and are identified where appropriate.

2.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

2.1.1 TSS Evaluation Activity

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

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

[ST] Section 7.1.1 states that only one authorization factor is supported, a user-supplied password composed of at least 10 characters and up to a maximum of 64 characters. This user-supplied password may contain upper case letters, lower case letters, numbers, special characters, or any other 8-bit ASCII value. These 8-bit values can be any of the 8-bit ASCII values, including special characters. While also supporting ATA and NVM standards, the TOE can accept any 8-bit value, or any combination of ones and zeroes making up the password.

Additional Assurance Activities regarding password-based authentication factors are contained in FCS_PCC_EXT.1.

[AGD] describes that the TOE accepts passwords length minimum 10 bytes up to 64 bytes. This is consistent with the description in the [ST].

The TOE does not support any other authorization factor other than the password.

2.1.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 11 lists and describes the TOE supported modes of operation – Security Mode Disabled, Security Mode Enabled and Exception Mode. For Security Mode disabled, security is not claimed at this state because the TOE has not activated Host Key Encryption. For Security Mode Enabled, only authorized user can access protected data after acquiring authorization. For Exception mode, the TOE is out of service to any host command except activity signal pin output signal.

[AGD] Sections 13 and 20 describe that the only authorization factor supported in ‘Security Mode Enabled’ mode is a user-supplied password composed of at least 10 characters and up to a maximum of 64 characters. This user-supplied password may contain upper case letters, lower case letters, numbers, special characters, or any other 8-bit ASCII value.

[AGD] Section 1 describes the Crypto Officer is responsible for configuring the product prior to field deployment by setting up the initial password.

[AGD] Sections 18 and 30 provide the instructions on initial configuration of the password authorization factor.

Additional Assurance Activities regarding password-based authentication factors are contained in FCS_PCC_EXT.1.

2.1.3 KMD Evaluation Activity

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

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

[ST] Section 7.1.1 describes that the BEV is not wrapped, but is directly derived from authorization factor using PBKDF.

[ST] Section 7.1.1 describes that the authorization factor is being provided as a string of any 8-bit values. [ST] Section 7.1.1 describes that the TOE uses PBKDF2 to condition the authorization factor and obtain BEV. [ST] Section 7.1.3 describes that the TOE uses the SP800-132 PBKDF Function using HMAC-SHA-256. The PBKDF Function uses 1000 iterations to output a 256-bit key. [ST] does not describe any processing for the authorization factor before key derivation is performed, [ST] Section 7.1.1 states that the authorization factor is directly input into PBKDF function.

2.1.4 Test Evaluation Activity

The password authorization factor is tested in FCS_PCC_EXT.1.

The evaluator shall also perform the following tests:

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

Test Objective	The TOE only supports one authorization factor. Test not applicable.
Test Steps Performed	N/A
Test Result	Pass

2.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

2.2.1 TSS Evaluation Activity

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

[ST] Section 7.1.1 states that a password is the only authorization factor used to gain access to user data after the TOE resumes from a Compliant power saving state: D3.

[ST] Section 7.1.1 describes that the TOE supports only one authorization factor.

2.2.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 6 describes how the TOE enforces operation in power states D0 (fully powered on) and D3 (fully powered off). A password is required to enter D0 state from a power cycle. [AGD] Section 6 describes that only power state D0 allows a user to access plaintext data in User disk after providing the correct authorization factor.

2.2.3 KMD Evaluation Activity

None.

2.2.4 Test Evaluation Activity

The evaluator shall perform the following test:

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

Test Number	1
Test Objective	Ensure the TOE will not grant access to data without valid authorization when transitioning from a compliant power-saving state. The TOE supports the transition from power saving state of D3(off) to D0(fully on).
Test Steps Performed	Powered-on, initialized the TOE using good password credential. Formatted the disk and copy arbitrary text files to the TOE. Powered-off the TOE. Powered-on the TOE, attempted to authenticate using an invalid password and verified that TSF data was not accessible. The evaluator then rebooted the TOE and used a valid password to successfully authenticate to the TOE and verified that TSF data was accessible. The evaluator performed this for both models in the TOE.
Test Result	Pass

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

2.3.1 TSS Evaluation Activity

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.

[ST] Section 7.1.2 describes the symmetric keys: AES-XTS 512-bit key that comprise the DEK that are generated by the TOE's internal DRBG. The DEK is protected using AES key wrap before storing them in the TOE's persistent NAND flash memory. [ST] Section 7.1.3 shows that AES-XTS key is a pair of 256-bit keys.

2.3.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 13 describes that the TOE encrypts all user data with AES-256 XTS. There are no other configuration options or support for different key sizes.

2.3.3 KMD Evaluation Activity

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.

[ST] Section 7.1.2 describes that the TOE uses two symmetric keys – BEV and DEK. BEV is derived from the Authorization submask and is used to unwrap the persistently stored DEK.

2.3.4 Test Evaluation Activity

None.

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

2.4.1 TSS Evaluation Activity

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

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

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

[ST] Sections 7.1.2 and 7.1.3 describe that DEK comprises two 256-bit keys that comprise one 512-bit key for XTS mode of operation and user data protection. This DEK is generated by the TOE's internal DRBG. The DEK was not generated outside of the TOE.

[ST] Section 7.1.3 describes how the TOE's internal Hash_DRBG (SHA-256) is used and how it functions. This section describes that the noise source for entropy is provided by a NAND flash threshold voltage raw noise source. The noise source provides a minimum entropy of 2.5-bits per 8-bit block. The conditioning component of the TOE provides DRBG with a 256-bit blocks of full entropy. DRBG is seeded with 265 bits of entropy and 128 bit nonce. [ST] Section 7.1.2 describes that to generate 512-bit DEK TOE places two calls to a DRBG and uses unmodified DRBG output.

2.4.2 Guidance Documentation Evaluation Activity

None.

2.4.3 KMD Evaluation Activity

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.

[ST] Section 7.1.2 describes that the DEK is generated inside the TOE and is stored inside the TOE in wrapped state. [ST] Section 7.1.2 states that once the TOE derives the KWK (BEV) as part of the authorization process, it is used to unwrap the DEK. [ST] Section 7.1.3 describes that the TOE uses AES Key Wrap as specified in ISO/IEC 18033-3 and NISP SP 800-38F.

2.4.4 Test Evaluation Activity

The evaluator shall perform the following tests:

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

Test Number	1
Test Objective	Generate a DEK using the RBG specified in FCS_RBG_EXT.1 with size of 256 bits.
Test Steps Performed	[ST] Section 6.1.1.4 selected "generate a DEK using the RBG as specified in FCS_RBG_EXT.1; 256 bits". While the administrator can generate keys from the TOE, the key generation parameters are not configurable. Key generation is tested in Steps 4 and 5 of FCS_CKM.4(b) Test 1.
Test Result	Pass

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

2.5.1 TSS Evaluation Activity

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.*

[ST] Section 7.1.2 describes that derived BEV and unwrapped DEK are being stored in the TOE SSD controller AES Register and SRAM volatile memory by the TOE. TOE directly manages that volatile memory, so no additional tracking of keys is needed.

[ST] Section 7.1.2 describes that the keys are removed from volatile memory when no longer needed, and when TOE transitions to the Compliant power-saving state supported by the TOE (D3 – off state).

[ST] Section 7.1.2 states that the TOE does not use temporary keys.

[ST] Section 7.1.2 states that the TOE is a Self-Encrypting Drive and does not interact with the host platform to perform the destruction or management of keys, as all keys are stored within the physical enclosure boundary of the drive.

2.5.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 13 states, “The administrator and system designer shall implement application techniques, safeguards, and/or procedures to assure that power is removed from the TOE, state D3 (cold), when the host system is left unattended. On removal of power, the TOE purges the DEK and enters a full-off state in less than 20 miliseconds.”

[AGD] Section 13 states, “The TOE is not dependent on the operational environment to perform DEK purging or memory clear operations. All operations that perform clear and purge operations, once triggered, operate independently from the host SATA or PCIe interface.”

2.5.3 KMD Evaluation Activity

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

[ST] Section 7.1.2 describes each key type, origin of each key and states that it is located in the TOE SRAM volatile memory. It states that BEV is derived from authorization factor using PBKDF, and DEK is generated by the TOE DRBG at provisioning or later obtained by unwrapping stored wrapped DEK. Unwrapped DEK can be stored in SRAM or in the Hardware AES register.

2.5.4 Test Evaluation Activity

None.

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

2.6.1 TSS Evaluation Activity

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

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

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

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

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

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

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

Key Description (name, type, size) [Required]	Password submask	BEV	Wrapped DEK	Unwrapped DEK
Storage Location (e.g. NAND flash, RAM, disk, EEPROM) [Required]	SRAM	SRAM	NAND memory	SRAM, AES Engine HW Register
Memory Type (Volatile/Non-Volatile) [Required]	Volatile Memory	Volatile Memory	Non- Volatile Memory	Volatile Memory
How key is introduced into Volatile memory [As applicable]	User input through TOE interface	PBKDF2 output	N/A	Output of DRBG, unwrapping of wrapped DEK
How key is overwritten [Required]	Overwrite of 0x00	Overwrite of 0x00	Block erase.	Overwrite by 0x55 or overwrite of 0x00 followed by 0x55 for SRAM, Overwrite by zeroes for hardware AES register.
Method of Key Destruction (must be consistent with FCS_CKM.4.1(b) selections) [Required]	Overwrite of 0x00	Overwrite of 0x00	Old index block is removed entirely by a block erase command.	Overwrite by 0x55 or overwrite of 0x00 for SRAM, Overwrite by zeroes for hardware AES register.
Memory Controller Description [Required] How the key is written to and read from memory (e.g., block level, cell level)?	Cell Level	Cell Level	Block Level	Cell Level
Memory Controller Description [Conditional] Does the TSS describe mechanisms where copies of the key could potentially exist (e.g., a copy with parity bits, a copy without parity bits, any mechanisms that are used for redundancy)?	Byte level random access	Byte level random access	Block level random access	Byte level random access
Does the TSS describe any configurations or circumstances that may not strictly conform to the key destruction requirement? If so, list here. [Conditional]	No	No	No	No

[ST] Section 7.1.2 lists the Cryptographic Key table and identifies the key sizes. This section also describes the destruction methods of 4 keys in the TOE keychain. The TSS describes the keys are protected using AES in key wrap mode and are stored in NAND flash memory.

[ST] Section 7.1.2, describes all keys in the TOE keychain stored in volatile memory are destroyed by overwrite with zeroes, overwrite with 0x55, or upon entering the only Compliant power-saving state supported by the TOE, D3". The TOE does not use wear leveling for this memory.

[ST] Section 6.1.1.7 FCS_CKM.4.1(b) contains an assignment for overwrite data pattern (0x55). This pattern does not contain CSP.

[ST] Section 7.1.2, describes wrapped DEK stored in non-volatile memory is removed using block erase command. This is not affected by wear-levelling. [ST] Section 7.1.2 describes that because of nature of block-erase read-verify procedure is not performed.

[ST] Section 7.1.2 states that the “TOE uses the ATA or NVM command set to interact with the host platform” and that the TOE utilizes block-erase commands for key destruction in NAND memory.

[ST] Section 7.2.1 describes that TOE uses 2 types of volatile memory: SRAM and DRAM. [ST] Section 7.1.2 states keys and key material are stored in the SRAM memory and TOE controller HW register (AES Register).

[ST] Section 7.1.2 describes that the BEV and password are removed from memory and zeroized right after unwrapping the DEK. Unwrapped DEK is removed from volatile memory when the device is powered-off, user is logged out, password is changed, or the TOE is zeroized. When the drive is unlocked, SATA models retain the DEK in SRAM and hardware AES Register, but the PCIe/NVMe drives destroy the DEK from SRAM after loading it into the hardware AES register.

[ST] Section 7.1.2 describes that the TOE wraps DEK from volatile memory and stores it in NAND memory at provisioning or when password change is requested.

[ST] Section 7.1.2 describes that the controller performs write operation as a byte level random access. During functional testing the evaluator observed that copies of keys or key fragments were not created in memory.

2.6.2 Guidance Documentation Evaluation Activity

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

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

Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

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

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

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

[AGD] Section 7 describes that the TOE design is based on a single ASIC controller which allows the TSF to initiate the zeroization service without delay to destroy key and key materials immediately via the ATA/NVM command or GPIO signal input specified in [SCG], regardless of ongoing operation such as garbage collection, wear-levelling, bad block management, or any other ATA command operation including TRIM. During the zeroize process, the TOE does not respond to host commands and the only activity is from the signal output indicator.

The evaluator establishes that there are no circumstances in which the TOE does not conform to the key destruction requirements.

2.6.3 KMD Evaluation Activity

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

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

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

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

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

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

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

This work unit is addressed in [ASE] evaluation. See [ASE] Section 12.8.

2.6.4 Test Evaluation Activity

For these tests the evaluator shall utilize appropriate development environment (e.g. a Virtual Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.
For destruction on wear-leveled memory, if a time period is required before is evaluator shall wait that amount of time after clearing the key in tests 2 and 3.

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

1. *Record the value of the key in the TOE subject to clearing.*
2. *Cause the TOE to perform a normal cryptographic processing with the key from Step #1.*
3. *Cause the TOE to clear the key.*
4. *Cause the TOE to stop the execution but not exit.*
5. *Cause the TOE to dump the entire memory of the TOE into a binary file.*
6. *Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.*
7. *Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.*

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

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

Test 2*: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:*

1. Record the value of the key in the TOE subject to clearing.

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

4. Search the non-volatile memory the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.

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

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

1. Record the storage location of the key in the TOE subject to clearing.

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

4. Read the storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

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

Test Number	1
Test Objective	Ensure the key is destroyed from volatile memory when no longer needed; i.e., KWK (BEV) should not be present when DEK unwrapping is complete, and DEK should not be present when device is in locked state. Keys should also be destroyed when TOE performs zeroization (DEK destruction when invoked by user). Keys and key fragments should not

	remain in volatile memory.
Test Steps Performed	This test was performed for multiple cases in which the evaluator expected that key destruction should occur to meet the SFR. These included the following points: • TOE is changing from Initialization to Login state (1) • TOE is changing from Login state to User state (2) • TOE is changing from User state to Login State (3) • TOE is changing from User state to Zeroized state (4) Debug versions of the TOE models were used for this test. (1) The evaluator powered on the TOE while the TOE was connected via JTAG and in an Uninitialized state, setting a break point to be at the end of the DRBG functionality (at the Login State). The evaluator initialized the TOE with a new password, and the breakpoint was triggered, allowing the plaintext password, KWK, DEK_A and DEK_B values to be read and recorded directly from TOE memory. The evaluator verified that DEK_A and DEK_B were 256 bits each. The evaluator allowed the TOE to continue execution of the TOE firmware code to arrive at the Login State of the TOE. At this point, the evaluator dumped the TOE memory (SRAM and AES memory), inspected the memory dumps for the presence of the values identified in earlier steps and verified that the values were not found. The evaluator repeated this process for the remaining TOE-state transitions identified as (2), (3) and (4) above. At each point, the evaluator verified that the KWK, DEK_A and DEK_B were not found. The evaluator performed this test on both the A and B models in the TOE.
Test Result	Pass

Test Number	2
Test Objective	Ensure keys are correctly destroyed from the non-volatile memory and they do not remain in non-volatile memory.
Test Steps Performed	The TOE does not destroy keys in non-volatile memory by overwrite, only by block erase.
Test Result	Pass

Test Number	3
Test Objective	Ensure keys are correctly destroyed from the non-volatile memory using a claimed overwrite method.
Test Steps Performed	The TOE does not destroy keys in non-volatile memory by overwrite, only by block erase.
Test Result	Pass

2.7 FCS_CKM.4(d) Cryptographic Key Destruction (Software TOE, 3rd Party Storage) (Selection-based for EE)

2.7.1 TSS Evaluation Activity

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

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

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

The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement. If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.

[ST] Section 7.1.2 describes that the TOE does not interact with the host platform to perform the destruction or management of keys, as all keys are stored within the physical enclosure of the drive. [ST] Section 7.1.2 states, “The KWK or BEV is never stored persistently and is therefore the only temporary key. The TOE does not store plaintext keys that are part of the keychain in persistent memory.”

[ST] Section 7.1.2 describes that the TOE uses ATA or NVM command set to interact with the host platform.

The evaluator concluded that this work unit is implicitly satisfied.

2.7.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 7 describes that the TOE design is based on a single ASIC controller which allows the TSF to initiate the zeroization service without delay to destroy key and key materials immediately via the ATA/NVM command or GPIO signal input specified in [SCG], regardless of ongoing operation such as garbage collection, wear-levelling, bad block management, or any other ATA command operation including TRIM. During the zeroize process, the TOE does not respond to host commands and the only activity is from the signal output indicator.

The evaluator establishes that there are no circumstances in which the TOE does not conform to the key destruction requirements.

2.7.3 KMD Evaluation Activity

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

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

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

The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement. If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.

[ST] Section 7.1.2 describes that the TOE does not interact with the host platform to perform the destruction or management of keys, as all keys are stored within the physical enclosure of the drive. [ST] Section 7.1.2 states, “The KWK or BEV is never stored persistently and is therefore the only temporary key. The TOE does not store plaintext keys that are part of the keychain in persistent memory.”

[ST] Section 7.1.2 describes that the TOE uses ATA or NVM command set to interact with the host platform.

The evaluator concludes that this work unit is implicitly satisfied.

2.7.4 Test Evaluation Activity

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.

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

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

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

1. Record the value of the key in the TOE subject to clearing.

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

4. Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.

5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for Use Case 1 test 1 above), and if a fragment is found in the repeated test then the test fails.

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

1. Record the logical storage location of the key in the TOE subject to clearing.

2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.

3. Cause the TOE to clear the key.

4. Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

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

Test Number	1
Test Objective	Test not applicable. The TOE does not use 3rd party storage to store key materials. See FCS_CKM.4(b) -Test 1 for applicable key destruction tests.
Test Steps Performed	N/A
Test Result	Pass

Test Number	2

Test Objective	This test is not applicable. The TOE does not use 3rd party storage to store key materials.
Test Steps Performed	N/A
Test Result	Pass

Test Number	3
Test Objective	This test is not applicable. The TOE does not use 3rd party storage to store key materials.
Test Steps Performed	N/A
Test Result	Pass

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

2.8.1 TSS Evaluation Activity

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.

[ST] Section 7.1.2 describes all keys are destroyed upon entering the only compliant power saving state supported by the TOE - D3 (off).

[ST] Section 7.1.2 describes that the user password and BEV are considered no longer needed and destroyed once DEK unwrapping is completed. DEK stored in non-volatile memory is needed until the TOE is re-provisioned, or given a password-reset command, which at that time, the wrapped DEK is zeroized and replaced with the value of a new wrapped DEK.

[ST] Section 7.1.2 contains a table summarizing events when key destruction occurs, together with the method used for key destruction in each case.

[ST] Section 7.1.2 states that the salt for key derivation, and the DEK, are considered needed for the whole length of the TOE operation and destroyed from volatile only when entering an off state.

[ST] Section 7.1.2 states that all keys and key material (salt) is also erased when secure erase is performed for the device.

2.8.2 Guidance Documentation Evaluation Activity

None.

2.8.3 KMD Evaluation Activity

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

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

[ST] Section 7.2.1 describes that the TOE uses two types of dedicated volatile memory: SRAM and DRAM. [ST] Section 7.2.1 states the unwrapped DEK is stored in SRAM memory and Hardware AES registry, that is located in SSD controller.

[ST] Section 7.1.2 describes that the BEV stored in volatile memory is considered no longer needed after successful unwrapping of the DEK. It states that the unwrapped DEK is considered needed and stored temporarily in SRAM volatile memory to load to the AES hardware register while the device is powered-on and running in User state. SATA models retain the DEK in SRAM, but PCIe/NVMe drives destroy it after loading it into the hardware register. It is always destroyed from volatile memory once the device is powered off, the user is logged out, or the TOE is zeroized. [ST] Section 7.1.2 states that the salts are destroyed from non-volatile memory only on device erasure and from volatile memory only when the TOE is powered off.

2.8.4 Test Evaluation Activity

None.

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

2.9.1 TSS Evaluation Activity

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

[ST] Section 7.1.2 describes that all keys and key material stored in plaintext in volatile memory are destroyed upon entering the only Compliant power-saving state supported by the TOE.

2.9.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 6 describes that the SSDs never receive warning of imminent power loss. This design of the TOE assures that there are no scenarios where an unexpected power loss can result in the TOE entering a non-compliant power state.

[AGD] Section 14 states, “the CO and system designers must implement host system application techniques, safeguards, and/or procedures that remove power from the TOE whenever the host platform is left unattended” when the “host system is unattended, or for example, in a lock-screen or sleep state”.

2.9.3 KMD Evaluation Activity

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

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM_EXT.6/FCS_CKM.4(d) for the destruction.

[ST] Section 7.2.1 describes that the TOE uses two types of volatile memory: SRAM and DRAM. [ST] Section 7.2.1 states the keys and key material are stored in SRAM memory and Hardware AES registry, which is located in the SSD controller.

[ST] Section 7.1.2 describes that BEV stored in volatile memory are considered no longer needed after successful unwrapping of DEK. It states that the unwrapped DEK is considered needed and stored temporarily in SRAM volatile memory to load to AES hardware register while the device is powered-on

and running in User state. SATA models retain the DEK in SRAM, but PCIe/NVMe drives destroy it after loading it into the hardware register. The DEK is always destroyed from volatile memory once the device is powered off, the user is logged out, or the TOE is zeroized.

[ST] Section 7.1.2 describes that the salts are destroyed from non-volatile memory only on device erasure and from volatile memory only when the TOE is powered off.

2.9.4 Test Evaluation Activity

None.

2.10 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

2.10.1 TSS Evaluation Activity

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.

[ST] Section 7.1.2 contains Table 9 summarizing events when key destruction occurs, together with method used for key destruction in each case. It describes how each of the keys are destroyed by "overwrite of 0x00 followed by 0x55" or block erase for keys in non-volatile memory. Unwrapped DEK in AES hardware register is overwritten with random pattern or with zeroes. This is consistent with the selections in the SFRs.

[ST] Section 7.1.2 describes that the TOE does not interact with the host platform to perform the destruction or management of the keys because all the keys are stored within the physical enclosure of the drive.

The evaluator determines that for all keys subject to destruction the TSS describes that they are destroyed according to one of the methods specified [ST] Sections 6.1.1.10, 6.1.1.6.

2.10.2 Guidance Documentation Evaluation Activity

None.

2.10.3 KMD Evaluation Activity

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.

[ST] Section 7.1.2, Table 9 describes the key destruction method for each key. [ST] Section 7.1.2 describes that the TOE does not interact with the host platform to perform the destruction or management of the keys because all the keys are stored within the physical enclosure of the drive.

The evaluator determined that for all keys subject to destruction, the TSS in [ST] Section 7.1.2 describes that they are destroyed according to one of the methods specified [ST] Section 6.1.1.10, by overwriting them with zeroes, random pattern or with 0x55 when in volatile memory or by block erase for non-volatile memory.

2.10.4 Test Evaluation Activity

None.

2.11 FCS_COP.1(a) Cryptographic Operations (Signature Verification)

2.11.1 TSS Evaluation Activity

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" vice "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).

[ST] Section 7.1.3 states that the TOE implements ECDSA using a P-384 curve.

[ST] Section 7.1.3 describes that the hash digest of the public key is hard-coded in the TOE. [ST] Section 7.4.4 elaborates on the storage location to state that this hash digest is "hard-coded in the ASIC controller one-time programmable fuse during the production stage by the vendor".

[ST] Section 7.4.4 describes the overall flow of the Trusted update process and describes the ECDSA signature verification process and how the TOE firmware is used to authenticate the data received from the operational environment. It describes that public key is supplied along with the digital signature and firmware image to the TOE via the ATA interface.

[ST] does not describe any other processing this is being used in digital signature verification, apart from checking the correctness of signature and comparing digest of public key to hardcoded value.

2.11.2 Guidance Documentation Evaluation Activity

None.

2.11.3 KMD Evaluation Activity

None.

2.11.4 Test Evaluation Activity

Each section below contains the tests the evaluators must perform for each type of digital signature scheme. Based on the assignments and selections in the requirement, the evaluators choose the specific activities that correspond to those selections.
It should be noted that for the schemes given below, there are no key generation/domain parameter generation testing requirements. This is because it is not anticipated that this functionality would be needed in the end device, since the functionality is limited to checking digital signatures in delivered updates. This means that the domain parameters should have already been generated and encapsulated in the hard drive firmware or on-board non-volatile storage. If key generation/domain parameter generation is required, the evaluation and validation scheme must be consulted to ensure the correct specification of the required evaluation activities and any additional components.
The following tests are conditional based upon the selections made within the SFR.
The following tests may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

ECDSA Algorithm Tests
ECDSA FIPS 186-4 Signature Verification Test
For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

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

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

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP Certificate A897 for ECDSA Signature Verification with P-384 curve and SHA2-384.
Test Result	Pass

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

2.12.1 TSS Evaluation Activity

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.

[ST] Section 7.1.3, Table 10, FCS_COP.1(b), states the module implements SHA-256 with block size 512 and SHA-384 with block size 1024.

[ST] Section 7.1.3, Table 10, FCS_COP.1(b), states the association of the hash function with other TSF cryptographic functions.

2.12.2 Guidance Documentation Evaluation Activity

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

[AGD] does not describe that the hash sizes are configurable, or system configuration (other than Enabling the Security mode) is required.

2.12.3 KMD Evaluation Activity

None.

2.12.4 Test Evaluation Activity

The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented test mode.
The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this cPP.
Short Messages Test Bit-oriented Mode

The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
Short Messages Test Byte-oriented Mode
The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
Selected Long Messages Test Bit-oriented Mode
*The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 99*i, where 1 ≤ i ≤ m. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
Selected Long Messages Test Byte-oriented Mode
*The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. For SHA-384 and SHA-512, the length of the i-th message is 1024 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
Pseudorandomly Generated Messages Test
This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of the NIST Secure Hash Algorithm Validation System (SHAVS) (https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Algorithm-Validation-Program/documents/shs/SHAVS.pdf). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP certificate A897 for SHA-384; C411 for SHA-256.
Test Result	Pass

2.13 FCS_COP.1(c) (AA) Cryptographic Operation (Keyed Hash Algorithm)

2.13.1 TSS Evaluation Activity

If HMAC was selected:

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

If CMAC was selected:

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

[ST] Section 7.1.3, Table 10, FCS_COP.1(c) states that the keyed hash algorithm as utilized in the SP800-132 PBKDF2, has the following attributes: 256-bit key, SHA-256 hash, 512-bit block size, with an output length of 256-bits.

2.13.2 Guidance Documentation Evaluation Activity

None.

2.13.3 KMD Evaluation Activity

None.

2.13.4 Test Evaluation Activity

If HMAC was selected: *For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key using a known good implementation.* *If CMAC was selected:* *For each of the supported parameter sets, the evaluator shall compose at least 15 sets of test data. Each set shall consist of a key and message data. The test data shall include messages of different lengths, some with partial blocks as the last block and some with full blocks as the last block. The test data keys shall include cases for which subkey K1 is generated both with and without using the irreducible polynomial R_b, as well as cases for which subkey K2 is generated from K1 both with and without using the irreducible polynomial R_b. (The subkey generation and polynomial R_b are as defined in SP800-38E.) The evaluator shall have the TSF generate CMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating CMAC tags with the same key using a known good implementation.*

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP certificate A897 for HMAC-SHA2-256.
Test Result	Pass

2.14 FCS_COP.1(c) (EE) Cryptographic Operation (Message Authentication)

2.14.1 TSS Evaluation Activity

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

[ST] Section 7.1.3, Table 10, FCS_COP.1(c) states that the keyed hash algorithm as utilized in the SP800-132 PBKDF2, has the following attributes: 256-bit key, SHA-256 hash, 512-bit block size, with an output length of 256-bits.

2.14.2 Guidance Documentation Evaluation Activity

None.

2.14.3 KMD Evaluation Activity

None.

2.14.4 Test Evaluation Activity

If HMAC was selected: *For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key using a known good implementation.* *If CMAC was selected:* *For each of the supported parameter sets, the evaluator shall compose at least 15 sets of test data. Each set shall consist of a key and message data. The test data shall include messages of different lengths, some with partial blocks as the last block and some with full blocks as the last block. The test data keys shall include cases for which subkey K1 is generated both with and without using the irreducible polynomial R_b, as well as cases for which subkey K2 is generated from K1 both with and without using the irreducible polynomial R_b. (The subkey generation and polynomial R_b are as defined in SP800-38E.) The evaluator shall have the TSF generate CMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating CMAC tags with the same key using a known good implementation.*

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP certificate A897 for HMAC-SHA2-256.
Test Result	Pass

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

2.15.1 TSS Evaluation Activity

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.

[ST] Section 7.1.3, Table 10, FCS_COP.1(d) states that AES Key Wrap operations use a 256-bit key and key wrap mode as specified in ISO/IEC 18033-3 NIST SP 800-38F.

2.15.2 Guidance Documentation Evaluation Activity

None.

2.15.3 KMD Evaluation Activity

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.

[ST] Section 7.1.3, Table 10, FCS_COP.1(d) states that AES Key Wrap operations use a 256-bit key and key wrap mode as specified in ISO/IEC 18033-3 NIST SP 800-38F. Key wrapping occurs when a password change command is successfully performed or at device provisioning.

2.15.4 Test Evaluation Activity

None.

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

2.16.1 TSS Evaluation Activity

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

[ST] Section 7.1.3, Table 10, FCS_COP.1(f) states that the user data is encrypted using AES in XTS mode, using a pair of 256-bit keys with AES-256 underlying encryption/decryption algorithms.

2.16.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 13 describes that the TOE encrypts all user data with AES-256 XTS. There are no other configuration options or support for different key sizes.

2.16.3 KMD Evaluation Activity

None.

2.16.4 Test Evaluation Activity

The following tests are conditional based upon the selections made in the SFR.
AES-CBC Tests
For the AES-CBC tests described below, the plaintext, ciphertext, and IV values shall consist of 128-bit blocks. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known-good implementation.
These tests are intended to be equivalent to those described in NIST's AES Algorithm Validation Suite (AESAVS) (http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf). Known answer values tailored to exercise the AES-CBC implementation can be obtained using NIST's CAVS Algorithm Validation Tool or from NIST's ACPV service for automated algorithm tests (acvp.nist.gov), when available. It is not recommended that evaluators use values obtained from static sources such as the example NIST's AES Known Answer Test Values from the AESAVS document, or use values not generated expressly to exercise the AES-CBC implementation.
AES-CBC Known Answer Tests
KAT-1 (GFSBox):
To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different plaintext values for each selected key size and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros.
To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different ciphertext values for each selected key size and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using a key value of all zeros and an IV of all zeros.
KAT-2 (KeySBox):

To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros.
To test the decrypt functionality of AES-CBC, the evaluator shall supply a set of five different key values for each selected key size and obtain the plaintext that results from AES-CBC decryption of an all-zeros ciphertext using the given key and an IV of all zeros.
KAT-3 (Variable Key):
To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of keys for each selected key size (as described below) and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using each key and an IV of all zeros.
Key i in each set shall have the leftmost i bits set to ones and the remaining bits to zeros, for values of i from 1 to the key size. The keys and corresponding ciphertext are listed in AESAVS, Appendix E.
To test the decrypt functionality of AES-CBC, the evaluator shall use the same keys as above to decrypt the ciphertext results from above. Each decryption should result in an all-zeros plaintext.
KAT-4 (Variable Text):
To test the encrypt functionality of AES-CBC, for each selected key size, the evaluator shall supply a set of 128-bit plaintext values (as described below) and obtain the ciphertext values that result from AES-CBC encryption of each plaintext value using a key of each size and IV consisting of all zeros.
Plaintext value i shall have the leftmost i bits set to ones and the remaining bits set to zeros, for values of i from 1 to 128. The plaintext values are listed in AESAVS, Appendix D.
To test the decrypt functionality of AES-CBC, for each selected key size, use the plaintext values from above as ciphertext input, and AES-CBC decrypt each ciphertext value using key of each size consisting of all zeros and an IV of all zeros.
AES-CBC Multi-Block Message Test
The evaluator shall test the encrypt functionality by encrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a plaintext message of length i blocks, and encrypt the message using AES-CBC. The resulting ciphertext values shall be compared to the results of encrypting the plaintext messages using a known good implementation.
The evaluator shall test the decrypt functionality by decrypting nine i-block messages for each selected key size, for 2 ≤ i ≤ 10. For each test, the evaluator shall supply a key, an IV, and a ciphertext message of length i blocks, and decrypt the message using AES-CBC. The resulting plaintext values shall be compared to the results of decrypting the ciphertext messages using a known good implementation.
AES-CBC Monte Carlo Tests
The evaluator shall test the encrypt functionality for each selected key size using 100 3-tuples of pseudo-random values for plaintext, IVs, and keys.
The evaluator shall supply a single 3-tuple of pseudo-random values for each selected key size. This 3-tuple of plaintext, IV, and key is provided as input to the below algorithm to generate the remaining 99 3-tuples, and to run each 3-tuple through 1000 iterations of AES-CBC encryption.

```
# Input: PT, IV, Key
Key[0] = Key
IV[0] = IV
PT[0] = PT
for i = 1 to 100 {
        Output Key[i], IV[i], PT[0]
for j = 1 to 1000 {
        if j == 1 {
CT[1] = AES-CBC-Encrypt(Key[i], IV[i], PT[1])
PT[2] = IV[i]
        } else {
CT[j] = AES-CBC-Encrypt(Key[i], PT[j])
PT[j+1] = CT[j-1]
        }
}
        Output CT[1000]
```

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

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

The ciphertext computed in the 1000th iteration (CT[1000]) is the result for each of the 100 3-tuples for each selected key size. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

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

AES-GCM Test

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

128 bit and 256 bit keys

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

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

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

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

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

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

XTS-AES Test

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

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

***Three data unit (i.e., plaintext) lengths**. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.*

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

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

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

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP certificate C448 for AES-XTS 256-bit.
Test Result	Pass

2.17 FCS_KDF_EXT.1 Cryptographic Key Derivation

2.17.1 TSS Evaluation Activity

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 SP800-132.

[ST] Section 7.1.3, Table 10 FCS_KDF_EXT.1 describes the PBKDF Function as using HMAC-SHA-256 and uses 1,000 iterations to output a 256-bit key according to SP 800-132.

2.17.2 Guidance Documentation Evaluation Activity

None.

2.17.3 KMD Evaluation Activity

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.

[ST] Section 7.1.2 describes that the BEV is derived using the PBKDF2 function.

[ST] Section 7.1.3, Table 10 FCS_KDF_EXT.1 describes the PBKDF2 function as using HMAC-SHA-256 and uses 1,000 iterations to output a 256-bit key according to SP 800-132.

2.17.4 Test Evaluation Activity

None.

2.18 FCS_KYC_EXT.1 Key Chaining (Initiator)

2.18.1 TSS Evaluation Activity

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

[ST] Section 7.1.2, Table 8: Keychain states the BEV size of 256 bits, in support of AES-256.

2.18.2 Guidance Documentation Evaluation Activity

None.

2.18.3 KMD Evaluation Activity

The evaluator shall examine the KMD describes a high level description of the key hierarchy for all authorizations methods selected in FCS_AFA_EXT.1 that are used to protect the BEV. The evaluator shall examine the KMD to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap or key derivation methods that meet FCS_COP.1(d) and FCS_KDF_EXT.1.

The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the effective strength of the BEV is maintained throughout the key chain.

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

[ST] Section 7.1.1 describes that the TOE supports only one authorization factor - password submask. This password is used in the PBKDF2 function to produce the BEV, in a way that satisfies FCS_KDF_EXT.1.

[ST] Section 7.1.2 describes that the salt used in the BEV derivation is produced by the TOE DRBG.

[ST] Section 7.1.2 describes that the authorization factor and salt are fed directly to the PKDF2 function to produce the BEV so that the effective strength and security of the BEV is maintained.

[ST] Section 7.1.2 describes that the BEV security strength is 256 bits.

2.18.4 Test Evaluation Activity

None.

2.19 FCS_KYC_EXT.2 Key Chaining (Recipient)

2.19.1 TSS Evaluation Activity

None.

2.19.2 Guidance Documentation Evaluation Activity

None.

2.19.3 KMD Evaluation Activity

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

The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM). This description must include a diagram illustrating the key

hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or knowledge of the BEV and the effective strength of the DEK is maintained throughout the Key Chain.

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

[ST] Section 7.1.2 describes that the keychain consists of the BEV and DEK, that is wrapped using AES-KW as defined in FCS_COP.1(d).

As the key chain consists only from BEV and DEK and BEV is used directly to perform unwrapping of the DEK, the keychain will not be compromised without BEV knowledge; therefore, the effective strength of the DEK is maintained.

2.19.4 Test Evaluation Activity

None.

2.20 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

2.20.1 TSS Evaluation Activity

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

[ST] Section 7.1.1 states that the minimum password length is 10 bytes, and the maximum limit is 64 bytes. This user-supplied password may contain any other 8-bit value. The password-based authorization factor is validated after the password is run through the PBKDF2 function to condition the password. The user-supplied password is fed into PBKDF2 directly.

[ST] Section 7.1.3, Table 10, “FCS_KDF_EXT.1” describes the PBKDF function using HMAC-SHA-256 with 1000 iterations and outputting a 256-bit key. The evaluator verified that this is consistent and conformant to FCS_PCC_EXT.1.1 as instantiated in [ST] Section 6.1.1.19.

2.20.2 Guidance Documentation Evaluation Activity

None.

2.20.3 KMD Evaluation Activity

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

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

[ST] Section 7.1.3 describes that the BEV is created using PBKDF2 function with HMAC-SHA-256, so the key size of 256 bits matches the selection in the [ST] Section 6.1.1.19 FCS_PCC_EXT.1.

[ST] Section 7.1.1 states that user input is passed to PBKDF2 function unmodified and without encoding processing.

[ST] Section 7.1.3. describes settings for the PBKDF2 algorithm as: 256-bit key, SHA-256 hash, 512-bit block size, with an output length of 256-bits, so the function output is of the same length as function input and of the same length as BEV. This description is consistent with selections performed in the SFRs.

2.20.4 Test Evaluation Activity

The evaluator shall also perform the following tests:

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

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

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

Test Number	1
Test Objective	Verify the TOE supports a minimum password/passphrase length of 64 characters.
Test Steps Performed	This test was performed on Model A and Model B. The evaluator changed the user password for the TOE to a 64-character password, then attempted to authenticate to the TOE using the old password and verified that the attempt was unsuccessful. The evaluator then successfully authenticated to the TOE using the new 64-character password.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE does not support a maximum password/passphrase length of more than 64 characters.
Test Steps Performed	This test was performed on Model A and Model B. Continuing from Test 1 above, the Evaluator connected the TOE to the SATA traffic analyzer and authenticated to the TOE then started the SATA traffic capture. The evaluator attempted to change the user password with a 65-character password and verified that the attempt was unsuccessful.
Test Result	Pass

Test Number	3
Test Objective	Verify all 8-bit values are supported by the TOE as valid password input.
Test Steps Performed	This test was performed on Model A and Model B. Continuing from Test 2 above, the Evaluator attempted to change the user password to a 64-character password which contained bytes from 00 to 3F and verified that the attempt was successful and verified that

	authenticating to the TOE using the new password was successful. The evaluator repeated these steps using byte ranges of 40 to 7F, 80 to BF, and C0 to FF.
Test Result	Pass

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

2.21.1 TSS Evaluation Activity

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.

RBG services for the TOE are provided by the TOE's Internal DRBG and no other sources.

[ST] Section 7.1.3 states that the TOE's internal DRBG uses a physical noise source consisting of NAND flash threshold voltage noise and each noise source provides a minimum entropy of 2.5-bits per 8-bit block. The TOE provides 1024-bits of conditioning input by concatenating 128 samples of 8 bits noise source, and the conditioning component utilizes the SHA-256 function in order to uniformly distribute entropy and provide a full 256-bit entropy input to DRBG. The internal DRBG uses this 256-bit entropy input and 128-bit entropy generated nonce to generate the salt and other keys and key materials used for the TSF.

[ST] Section 7.1.3, Table 10 FCS_RBG_EXT.1 states hash DRBG uses SHA-256 according to SP 800-90A.

[ST] Section 7.1.3 identifies the usage of the TOE's internal DRBG to be used to generate salts, keys and key material used for the TSF.

2.21.2 Guidance Documentation Evaluation Activity

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.

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

[AGD] Section 30 states that the users do not need to configure the DRBG.

2.21.3 KMD Evaluation Activity

None.

2.21.4 Test Evaluation Activity

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

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

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

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

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

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

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

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

Test Number	1
Test Objective	Verify that cryptographic algorithms are implemented correctly.
Test Steps Performed	Satisfied by CAVP certificate C463 for Hash DRBG with SHA2-256 primitive.
Test Result	Pass

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

2.22.1 TSS Evaluation Activity

The evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generating using an RBG described in FCS_RBG_EXT.1 or by the Operational Environment. If external function is used for this purpose, the TSS should include the specific API that is called with inputs.

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

[ST] Sections 7.1.2 and 7.1.3 describe the TOE using its internal Hash_DRBG(SHA-256) as specified in FCS_RBG_EXT.1 to generate a 256-bit salt for use in the PBKDF operation.

The TOE does not use external functions to generate salts.

[ST] Section 7.1.3 also describes that the AES XTS tweak value is not randomly generated but instead utilizes the Data Unit Sequence Number of the NAND flash physical address information (bank * block * page) ranging between 0 and 4,294,967,296.

2.22.2 Guidance Documentation Evaluation Activity

None.

2.22.3 KMD Evaluation Activity

None.

2.22.4 Test Evaluation Activity

None.

2.23 FCS_VAL_EXT.1(AA) Validation

2.23.1 TSS Evaluation Activity

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

[ST] Section 7.1.1 states that only one authorization factor (i.e., password) supports validation.

The TOE does not use multiple submasks within the TOE, so only one submask is validated.

2.23.2 Guidance Documentation Evaluation Activity

(conditional) If the validation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established. *(conditional) If the validation functionality is specified by the ST author, the evaluator shall examine the operational guidance to ensure that it states the values that the TOE uses for limits regarding validation attempts.*

The validation functionality is not configurable.

[AGD] Section 22 describes the TOE limits validation attempts to 10 attempts. [AGD] Section 22 describes that to protect SSD from brute-force attacks, the module implements a Key Retry count or current password fail count (“N”). When the Key Retry count is greater than 10, the SSD will proceed with the zeroization process automatically.

2.23.3 KMD Evaluation Activity

None.

2.23.4 Test Evaluation Activity

The evaluator shall perform the following tests: ***Test 1:*** *The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any "lockout" period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.* ***Test 2:*** *For each validated authorization factor, ensure that when the user provides an incorrect authorization factor, the TOE prevents the BEV from being forwarded outside the TOE (e.g., to the EE).*

Test Number	1
Test Objective	Verify the TOE erases the cryptographic keys when 'failed attempts limit' of 10 attempts is reached.
Test Steps Performed	This test was performed on Model A and Model B. This test satisfies testing for both AA and EE. Continuing from FCS_PCC_EXT.1 – Test 3, the Evaluator power-cycled the TOE, input an incorrect password 11 times, and verified that after 11 attempts to login to the TOE, the TOE changed its state to Uninitialized and presented a full unallocated drive to the user without user data in it, indicating the TOE performed a zeroization.
Test Result	Pass

Test Number	2 (AA)
Test Objective	Verify that BEV is not being forwarded after wrong credentials are entered.
Test Steps Performed	This test satisfies testing for AA only. This test was performed on Model A and Model B. The test was performed using debug versions of the models. Continuing from TOE state at the end of FCS_CKM.4(b) - Test 1; the Evaluator initialized the TOE with a new password and captured the KWK (BEV) value. The Evaluator then authenticated to the TOE using an incorrect password and verified in the SATA data capture that KEK(BEV) fragment values were not found in the capture.
Test Result	Pass

2.24 FCS_VAL_EXT.1(EE) Validation

2.24.1 TSS Evaluation Activity

The evaluator shall examine the TSS to determine which authorization factors support validation. *The evaluator shall examine the TSS to review a high-level description if multiple submasks are used within the TOE, how the submasks are validated (e.g., each submask validated before combining, once combined validation takes place).* *The evaluator shall also examine the TSS to determine that a subset or all of the authorization factors identified in the SFR can be used to exit from a Compliant power saving state.*

[ST] Section 7.1.1 states that only one authorization factor (i.e., password) supports validation. The TSS describes that password-based authorization factor is not conditioned. [ST] Section 7.1.1 describes that the TOE validates the BEV by comparing a SHA-384 hash of the BEV to the stored hash. Upon successful validation the BEV is used to unwrap the DEK. The TOE does not use multiple submasks within the TOE.

[ST] Section 7.1.1 states password-based authorization factor is always required when resuming from the only compliant power-saving state.

2.24.2 Guidance Documentation Evaluation Activity

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

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

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

The validation functionality is not configurable.

[AGD] Section 22 describes that the TOE limits validation attempts to 10 attempts. [AGD] Section 22 describes that to protect SSD from brute-force attacks, the module implements a Key Retry count or current password fail count (“N”). When the Key Retry count is greater than 10, the SSD will proceed with the zeroization process automatically.

The password-based authorization factor is always required when resuming from the only compliant power-saving state, D3 (off), supported by the TOE.

2.24.3 KMD Evaluation Activity

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

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

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

[ST] Section 7.1.1 describes that the TOE validates the BEV by comparing SHA-384 hash of the BEV to the stored hash. Upon successful validation the BEV is used to unwrap the DEK. This way the BEV is validated without exposing material that might compromise the submasks.

[ST] Section 7.1.1 describes that the TOE maintains a persistent counter for failed validation attempts.

2.24.4 Test Evaluation Activity

The evaluator shall perform the following tests:

***Test 1:** The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization*

factors in consecutive attempts to access the protected data. If the limit mechanism includes any "lockout" period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.

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

Test Number	1
Test Objective	Verify the TOE erases the cryptographic keys when 'failed attempts limit' of 10 attempts is reached.
Test Steps Performed	This test is satisfied by the testing performed for FCS_VAL_EXT.1(AA) - Test 1.
Test Result	Pass

Test Number	2 (EE)
Test Objective	Verify the TOE requires a valid password when resuming from a power saving state.
Test Steps Performed	This test satisfies testing for EE only. This test is satisfied by the testing performed for FCS_AFA_EXT.2.
Test Result	Pass

2.25 FDP_DSK_EXT.1 Protection of Data on Disk

2.25.1 TSS Evaluation Activity

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

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

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

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

[ST] Section 7.2.1 states the TOE uses a single ASIC controller with a built-in hardware AES encryption engine which controls all memory components directly.

[ST] Section 7.2.1 describes how the TOE performs encryption operations when data are being written to disk and decryption operation when data are being read from the drive. The description is sufficiently detailed to describe how data is being handled and at what moment encryption/decryption is performed.

[ST] Section 7.2.1 states “The TOE receives data from the host platform or microcontroller through the SATA/PCIe connections, to which data is then passed through the AES encryption engine before writing the encrypted data to the TOE’s NAND non-volatile memory. If there are multiple drives installed in the host system, it is up to the user to ensure that data is being sent to the TOE for encryption. If the TOE is the only storage medium in the host system, there should be no need for user intervention to ensure that data is sent to the TOE for encryption.”

During functional testing evaluator confirmed that the encryption is being performed transparently during read\write operations on disk, using TOE AES encryption engine located in the TOE controller, as described in the TSS.

No cryptographic functions are provided by the Operational Environment.

[ST] Sections 7.4.3 and 7.2.1 describe the after initialization (boot) the TOE performs self-tests and enters login state. At login state the TOE presents the read-only drive to the user. After the user is authenticated the TOE can write and read encrypted data to and from the drive.

[ST] Section 7.2.1 describes that all read and write operations to the disk are encrypted. Disk operates this way after it is provisioned and authenticated. [ST] Section 7.2.1 describes that all host accessible areas of the drive, including MBR and Partition table are encrypted.

[ST] Section 7.1.3, Table 10 lists the cryptographic operations that the TOE itself implements and utilizes.

The TOE is a solid state self-encrypting drive and does not support encryption of multiple disks.

2.25.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 30 describes the security rules, procedural and preparatory steps needed to enable the FDE functionality.

[SCG] Sections 2, 4, 5, and 6 describe configuration of the TOE to ensure that FDE functionality is enabled. [SCG] Section 2 describes that the TOE security functions can be controlled by using the available command path: ATA security command, ATA vendor command, NVM admin command.

The procedural steps are sufficient to ensure that all hard drive devices will be encrypted when encryption is enabled.

2.25.3 KMD Evaluation Activity

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

showing the main components within the data path and that it clearly identifies the data encryption engine.

The evaluator shall verify the KMD provides sufficient instructions for all platforms to ensure that when the user enables encryption, the product encrypts all hard storage devices. The evaluator shall verify that the KMD describes the data flow from the device's host interface to the device's persistent media storing the data. The evaluator shall verify that the KMD provides information on those conditions in which the data bypasses the data encryption engine (e.g. read-write operations to an unencrypted Master Boot Record area).

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

[ST] Section 1.4.1 contains a diagram (Figure 1) that shows the logical composition of the TOE, indicating the TOE interfaces, integrated controller, encryption engine, volatile and non-volatile memory blocks. This diagram illustrates that the encryption engine is embedded into a controller and is on the data path between the TOE SATA/PCIe external interface and non-volatile storage. [ST] Section 7.2.1 states that all data requests are processed through the TOE controller and TOE encryption engine.

[ST] Section 7.2.1 describes that all read and write operations to user accessible areas of the disk go through the encryption engine. [ST] Section 7.1.2 describes that wrapped DEK is stored in user-inaccessible part of the drive.

[ST] Section 7.2.1 describes that all host accessible areas of the drive, including MBR and Partition table are encrypted.

[ST] Section 7.1.2 states that a new keychain is generated when the device is provisioned, and an initial password is set up.

[AGD] Section 5 describes that the product enables encryption when a user password is set. At this point the contents of the hard drive is erased.

[ST] Section 7.1.2 describes that after power-on the modules enter a Login state or unauthenticated state and present only a read-only shadow drive that does not accept user data.

During functional testing, the evaluator verified that it is impossible to write user data to the drive prior to initialization of encryption engine and successful authentication, as the TOE does not present user-writable space before and during TOE initialization, and presents only a read-only shadow drive after initialization.

2.25.4 Test Evaluation Activity

The evaluator shall perform the following tests:

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

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

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

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

Test 3: *Verify data is encrypted:*

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

Test Number	1, 2 and 3
Test Objective	Initialize the drive, write random data to address 0, max LBA, and approximately the middle LBA. Change the cryptographic key and verify that the previously written data is no longer readable.
Test Steps Performed	This test was repeated for both Model A and Model B. The Evaluator initialized and authenticated to the TOE, generated a 1024 KB large random set of data using a non-TOE entity, wrote this random data to a specific sector on the TOE, power cycled the TOE, authenticated to the TOE again, read the TOE data at three specific LBA addressed and verified that the expected data was present. The evaluator then invoked the DEK erase function of the TOE using the Zeroize Disk command, reinitialized the TOE, authenticated to the TOE and read data from the TOE to verify that the known random data set that was written to the TOE in the previous steps, was not present.
Test Result	Pass

2.26 FMT_MOF.1 Management of Functions Behavior

2.26.1 TSS Evaluation Activity

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

[ST] Section 7.4.2 describes how the compliant power saving states: D3 are managed. This section also states that the TOE does not allow administrators or users to manage or configure the Compliant power saving states supported by the TOE.

2.26.2 Guidance Documentation Evaluation Activity

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

The TOE does not allow the administrators or users to manage or configure the Compliant power saving states behaviour.

2.26.3 KMD Evaluation Activity

None.

2.26.4 Test Evaluation Activity

The evaluator shall perform the following tests: ***Test 1:*** *The evaluator presents a privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior and properties are allowed.* ***Test 2:*** *The evaluator presents a non-privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior are not allowed.*

Test Number	1
Test Objective	Test not applicable. The TOE does not support management of Compliant power saving states.
Test Steps Performed	N/A
Test Result	Pass

Test Number	2
Test Objective	Test not applicable. The TOE does not support management of Compliant power saving states.
Test Steps Performed	N/A
Test Result	Pass

2.27 FMT_SMF.1(AA) Specification of Management Functions

2.27.1 TSS Evaluation Activity

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to change the DEK. *If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to cryptographically erase the DEK.* *If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the methods by which users may change the set of all authorization factor values supported.* *If item d) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.* *If item e) is selected in FMT_SMF.1.1: If power saving states can be managed, the evaluator shall ensure that the TSS describes how this is performed, including how the TOE supports disabling certain power saving states if more than one are supported. If additional management functions are claimed in the ST, the evaluator shall ensure the TSS describes the additional functions.*

For items (a) and (b):

The TOE incorporates EE in the TOE boundary, so forwarding of requests is implicitly satisfied by the TOE functionality, so this forwarding functionally was not included in the [ST].

For item (c):

[ST] Section 7.3.1 states that “Changing to a new password is only available in the User State after acquiring authorization, which requires providing the correct current password first before changing”.It also states that password can be changed using ATA/NVM command as described in the [SCG] or using admin software tool. [ST] Section 7.1.2 states that “A new KWK is created via the PBKDF function and used to wrap the existing plaintext DEK”.

For item (d):

[ST] Section 7.3.1 states that the “Initiate Firmware update process is invoked by using a dedicated firmware update tool which is released via the vendor support site”.

For item (e):

Item e) is not selected in [ST] Section FMT_SMF.1.1 (AA).

2.27.2 Guidance Documentation Evaluation Activity

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

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

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

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

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

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

For items (a) and (b):

The TOE incorporates EE in the TOE boundary, so forwarding of requests is implicitly satisfied by the TOE functionality, no user interaction is required.

For item (c):

[AGD] Section 18 states, “In order to change to a new Host Key, original Host Key is required to fills in first to enter User State in which authorized user can change to new Host Key by using “Change Host Key” service. When “Change Host Key” service performs successfully in User State, the TOE re-appears to host in Login State. Now only new Host Key is valid for authentication, and old Host Key is neither usable nor traceable.”

For item (d):

[AGD] Section 23 describes how to initiate TOE firmware/software updates using the Firmware update tool. This section describes where a user can get the firmware update tool, the signature validation process, steps to update the firmware/software and error messages for failed situations.

For item (e):

Not selected.

[AGD] Section 8 states, “The TOE contains no mechanism to allow export or recovery of keys or key materials.”

[AGD] Section 6 describes that the TOE enforces operation in power states D0 (fully on) and D3 (cold). A password is required to enter D0 state from a power cycle. [AGD] Section 6 describes that only power state D0 allows a user to access plaintext data in User disk.

2.27.3 KMD Evaluation Activity

None.

2.27.4 Test Evaluation Activity

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

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

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

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

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

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

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

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

***Test 5** (conditional): If the TOE provides the ability to disable the use of one or more power saving states, the evaluator shall devise a test that enables all supported power saving states and demonstrates that the*

TOE can enter into each of these states. The evaluator shall then disable the supported power saving states one by one, repeating the same set of actions that were performed at the start of the test, and observe each time that when a power saving state is configured to no longer be used, none of the behavior causes the disabled state to be entered.

Test Number	1
Test Objective	Verify the TOE has the functionality to forward a command to the EE to change and cryptographically erase the DEK. Verify the TOE supports changing authorization factor.
Test Steps Performed	The TOE incorporates EE in the TOE boundary, so forwarding of requests is implicitly satisfied by the TOE functionality. Authorization factor change testing is satisfied by the testing performed in FCS_PCC_EXT.1 – Test 1.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE has the functionality to initiate TOE firmware/software updates.
Test Steps Performed	This test is satisfied by the testing performed in FPT_TUD_EXT.1 - Test 1.
Test Result	Pass

Test Number	3
Test Objective	Test not applicable; the TOE does not support key recovery procedures.
Test Steps Performed	N/A
Test Result	Pass

Test Number	4
Test Objective	Test not applicable; the TOE does not support power saving states that are entered by certain events.
Test Steps Performed	N/A
Test Result	Pass

Test Number	5
Test Objective	Test not applicable; the TOE does not support managing power saving states.
Test Steps Performed	N/A
Test Result	Pass

2.28 FMT_SMF.1(EE) Specification of Management Functions

2.28.1 TSS Evaluation Activity

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

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

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

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

For item (a):

[ST] Section 7.3.1 describes that the TOE changes the DEK only when erasing the DEK using a zeroize command.

For item (b):

[ST] Section 7.3.1 describes how the TOE cryptographically erases the DEK by issuing a zeroize command: by triggering the hardware Secure Erase signal, sending Zeroize command, or exceeding the maximum password attempts count. [ST] Section 7.2.1 describes that the hardware signal is issued through GPIO pins.

For item (c):

[ST] Section 7.3.1 describes the process to initiate a TOE firmware update using a vendor-supported dedicated firmware update tool.

For item (d):

[ST] Section 6.1.3.2 contains assignment for “zeroize user data”. [ST] Section 7.3.1 describes zeroization or keys and user data is triggered by hardware signal or sending zeroize command or by exceeding maximum login attempts.

2.28.2 Guidance Documentation Evaluation Activity

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

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

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

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

For items (a):

[AGD] Section 7 describes that the TOE only changes the DEK by cryptographically erasing the DEK and removing all user data by triggering the hardware Secure Erase signal, sending Zeroize command, or exceeding the maximum password attempts count.

[AGD] Section 7 describes that the TOE design is based on a single ASIC controller which allows ASIC controller initiate zeroize. After completing the zeroize process, the TOE will appear to host in Uninitialized State with all user data erased.

For item (c):

[AGD] Section 23 describes how to initiate TOE firmware/software updates using the Firmware update tool. This section describes where a user can get the firmware update tool, the signature validation process, steps to update the firmware/software and error messages for failed situations.

For item (d):

[ST] Section 6.1.3.2 contains assignment for “zeroize user data”. [AGD] Section 7 describes this functionality.

[AGD] Section 8 states “The TOE contains no mechanism to allow export or recovery of keys or key materials.”

2.28.3 KMD Evaluation Activity

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.

The TOE does not support import of a wrapped DEK, so this requirement is implicitly satisfied.

2.28.4 Test Evaluation Activity

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to change and cryptographically erase the DEK (effectively removing the ability to retrieve previous user data). *If item c) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.* *If item d) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.*

Test Number	1
Test Objective	Verify the TOE has the functionality to change [“item a)”] and cryptographically erase the DEK [“item b)”] (effectively removing the ability to retrieve previous user data). Verify that the TOE has the functionality to initiate TOE firmware/software updates [“item c)”]. Verify that the TOE has the functionality to zeroize user data [“item d)”].
Test Steps Performed	Testing for “item a)” and “item b)” is satisfied by the testing performed in FCS_CKM.4 - Test 1. Testing for “item c)” is satisfied by the testing performed in FPT_TUD_EXT.1 – Test 1. Testing for “item d)” is satisfied by the testing performed in FDP_DSK_EXT.1 – Test 1, 2 and 3.
Test Result	Pass

2.29 FMT_SMR.1 Security Roles

2.29.1 TSS Evaluation Activity

None.

2.29.2 Guidance Documentation Evaluation Activity

None.

2.29.3 KMD Evaluation Activity

None.

2.29.4 Test Evaluation Activity

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

2.30 FPT_FUA_EXT.1 Firmware Update Authentication

2.30.1 TSS Evaluation Activity

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.

[ST] Section 7.4.4 describes the Root of Trust for Update (RTU) for the TOE is the hash value of the public key, that is compared to the hash value of the key used in the update signature. This hash value is hard-coded in the TOE internal fusebox.

The TOE verifies that the firmware update is correctly signed using the attached public key and that hash value of the public key matches the hard-cored RTU value. If this is not correct the update is rejected.

[ST] Section 7.4.4 describes the original firmware is not stored and is erased after completing a successful firmware update on the TOE.

2.30.2 Guidance Documentation Evaluation Activity

None.

2.30.3 KMD Evaluation Activity

None.

2.30.4 Test Evaluation Activity

None.

2.31 FPT_KYP_EXT.1 Extended: Protection of Key and Key Material [TD0458[1]]

2.31.1 TSS Evaluation Activity

[1] The evaluation activities were modified by TD0458.

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

[ST] Section 7.4.1 states the only 1 key is stored persistently in non-volatile NAND flash which is wrapped DEK. It is protected using AES-KW.

2.31.2 Guidance Documentation Evaluation Activity

None.

2.31.3 KMD Evaluation Activity

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.

[ST] Section 7.1.2 states that the only key stored in non-volatile memory is the DEK and it is stored in a wrapped form using AES-KW. No keys are stored in non-volatile memory in plaintext form.

2.31.4 Test Evaluation Activity

None.

2.32 FPT_PWR_EXT.1 Power Saving States (AA)

2.32.1 TSS Evaluation Activity

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

[ST] Section 7.4.2 lists that the TOE supports the Compliant power saving state D3.

2.32.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 6 describes that the TOE enforces operation in power states D0 (fully on) and D3 (cold).

[AGD] Section 6 states that the TOE does not allow power saving state configurations, and that the TOE does not support additional power saving states.

2.32.3 KMD Evaluation Activity

None.

2.32.4 Test Evaluation Activity

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

Test Number	1
Test Objective	Verify the TOE is removing all key material from volatile memory when transitioning between power saving states.
Test Steps Performed	The TOE supports only D3 (power-off) state as a compliant power-saving state When the TOE enters power-off state the power is removed from volatile memory. Information stored in the volatile memory is destroyed when power is lost by the nature of the volatile memory. Test 1 in FCS_CKM.4(d) describes that if key destruction is performed by removing power to volatile memory additional testing is unnecessary. This activity is implicitly satisfied.
Test Result	Pass

2.33 FPT_PWR_EXT.1 Power Saving States (EE) [TD0460[2]][TD0464[3]]

2.33.1 TSS Evaluation Activity

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

[ST] Section 7.4.2 lists that the TOE supports the Compliant power saving state D3.

2.33.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 6 describes that the TOE enforces operation in power states D0 (fully on) and D3 (cold).

2.33.3 KMD Evaluation Activity

None.

2.33.4 Test Evaluation Activity

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.

Test Number	1
Test Objective	Verify the TOE is removing all key material from volatile memory when transitioning between power saving states.
Test Steps Performed	The TOE supports only D3 (power-off) state as a compliant power-saving state. When the TOE enters power-off state the power is removed from volatile memory. Information stored in the volatile memory is destroyed when power is lost by the nature of the volatile memory. Tests indicated by selections in FCS_CKM_EXT.6 are Test 1 in FCS_CKM.4(b) and Test 1 in FCS_CKM.4(d). Those tests state that if key destruction is performed by removing power to volatile memory additional testing is unnecessary. This activity is implicitly satisfied.
Test Result	Pass

[2] The SFR and Assurance Activity text are modified by TD0229.
[3] The SFR and Assurance Activity text are modified by TD0229.

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

2.34.1 TSS Evaluation Activity

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

[ST] Section 7.4.2 states the TOE enters this Compliant power saving state upon request from an authorized user and in the event of a system shutdown.

2.34.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 6 describes that the TOE enforces operation in power states D0 (fully on) and D3 (cold).

[AGD] Section 6 describes that the SSDs never receive warning of imminent power loss. This design of the TOE assures that there are no scenarios where an unexpected power loss can result in the TOE entering a non-compliant power state.

2.34.3 KMD Evaluation Activity

None.

2.34.4 Test Evaluation Activity

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

Test Number	1
Test Objective	Verify the TOE correctly enters power saving state (Power-Off).
Test Steps Performed	This test was performed on Model A and Model B. The evaluator initialized the TOE and authenticated to the TOE, then instructed the non-TOE Operating System to enter the sleep state. The Evaluator then resumed the OS from the sleep state and verified that the TOE was in the 'Login State' (which requires authentication to enter User State).
Test Result	Pass

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

2.35.1 TSS Evaluation Activity

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

[ST] Section 7.4.2 states the TOE enters this Compliant power saving state upon request from an authorized user and in the event of a system shutdown.

2.35.2 Guidance Documentation Evaluation Activity

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).

[AGD] Section 6 describes the TOE enforces operation in power states D0 (fully on) and D3 (cold).

[AGD] Section 13 states, “The administrator and system designer shall implement application techniques, safeguards, and/or procedures to assure that power is removed from the TOE, state D3 (cold), when the host system is left unattended. On removal of power, the TOE purges the DEK and enters a full-off state in less than 20 miliseconds.”

2.35.3 KMD Evaluation Activity

None.

2.35.4 Test Evaluation Activity

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.

Test Number	1
Test Objective	Verify the TOE correctly enters power saving state (Power-Off).
Test Steps Performed	This test was performed on Model A and Model B. The evaluator initialized the TOE and authenticated to the TOE, then instructed the non-TOE Operating System to enter the sleep state. The Evaluator then resumed the OS from the sleep state and verified that the TOE was in the ‘Login State’ (which requires authentication to enter User State).
Test Result	Pass

2.36 FPT_RBP_EXT.1 Rollback Protection

2.36.1 TSS Evaluation Activity

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.

[ST] Section 7.4.4 describes the process of what the TOE verifies before updating the firmware image using the vendor’s firmware update tool.

[ST] Section 7.4.4 states a high-level description of the types of error codes provided and when these errors would be triggered. These include the following:

- “Invalid firmware image” when ECDSA with tagged digital signature fails for firmware image integrity;
- “Not user state” when the TOE is not in user state and;
- “Backward update error” when user attempts firmware image rollback.

2.36.2 Guidance Documentation Evaluation Activity

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

[AGD] Section 23 describes what each error code stands for and how the user should interpret the error codes: Backward update error, Invalid FW image.

[AGD] Section 23 states, “Firmware update can be proceeded only to newer firmware version which is using higher numbering than older version.”

2.36.3 KMD Evaluation Activity

None.

2.36.4 Test Evaluation Activity

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.*

Test Number	1
Test Objective	Verify the TOE will not accept a firmware upgrade with a lower version.
Test Steps Performed	This test was performed on Model A and Model B. Continuing with the TOE state from FPT_TUD_EXT.1, Step 10, the Evaluator started a SATA traffic capture, attempted to perform a firmware update with a firmware version number that was lower than that which was currently installed, and verified using the SATA data capture that the attempt failed.
Test Result	Pass

2.37 FPT_TST_EXT.1 Extended: TSF Testing

2.37.1 TSS Evaluation Activity

The evaluator shall verify that the TSS describes the known-answer self-tests for cryptographic functions. *The evaluator shall verify that the TSS describes, for some set of 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, for each of these functions, the method by which the TOE verifies the correct operation of the function. The evaluator shall verify that the TSF data are appropriate for TSF Testing. For example, more than blocks are tested for AES in CBC mode, output of AES in GCM mode is tested without truncation, or 512-bit key is used for testing HMAC-SHA-512.* *If FCS_RBG_EXT.1 is implemented by the TOE and according to NIST SP 800-90, the evaluator shall verify that the TSS describes health tests that are consistent with section 11.3 of NIST SP 800-90.* *If any FCS_COP functions are implemented by the TOE, the TSS shall describe the known-answer self-tests for those functions.* *The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe,*

for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall determine that all of the identified functions/components are adequately tested on start-up.

[ST] Section 7.4.3, Table 11: Self-Tests lists self-tests that are performed by the TOE and a brief description of the tests including the cryptographic tests and key sizes.

[ST] Section 7.4.3 states that the DRBG health check is performed per SP 800-90A. The DRBG Health test performs the KAT (Known Answer Test) by combining Instantiation, Reseed, and Generate. These health tests are consistent with section 11.3 of NIST SP 800-90A.

[ST] Section 7.4.3 Table 11 lists corresponding known-answer-tests for FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), and FCS_COP.1(f).

[ST] Section 7.4.3 lists which self-tests are performed on start-up and before using. [ST] Section 1.4.2.4 states, “The TOE protects itself by running a suite of self-tests at power-up and “before using” certain functions, authenticating firmware and by not providing any mechanism to export any key values.”

[ST] Section 7.4.3 describes Firmware Image Verification using hash calculation using SHA-384 and comparing that to the hash value, stored on a NAND storage.

2.37.2 Guidance Documentation Evaluation Activity

None.

2.37.3 KMD Evaluation Activity

None.

KMD Evaluation Results

2.37.4 Test Evaluation Activity

None.

2.38 FPT_TUD_EXT.1 Trusted Update

2.38.1 TSS Evaluation Activity

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

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

[ST] Section 7.4.4 states the vendor releases a firmware update tool to each authorized user who authenticated their identity by logging into the vendor support site.

[ST] Section 7.4.4 states the signature verification process utilizes ECDSA using a P-384 curve.

[ST] Section 7.4.4 states the firmware update is signed and distributed by the manufacturer using support website.

[ST] Section 7.4.4 states the public key used in the Trusted Update mechanism is “hard-coded in the ASIC controller fusebox”.

[ST] Section 7.4.4 describes the TOE protects and maintains the TOE update credentials. The TOE verifies the authority of the user by allowing the firmware update process only in the User state after logging in with the correct password.

The Operational Environment does not perform the signature verification, the TOE performs the signature verification before accepting the updated firmware using security functions in the TOE firmware.

2.38.2 Guidance Documentation Evaluation Activity

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.

[AGD] Section 23 describes how consumers of the TOE obtain the vendor provided update to the TOE. This is achieved via Novachips support site with a unique username and password.

[AGD] Section 23 describes how to initiate TOE firmware/software updates using the Firmware update tool. This section describes where a user can get the firmware update tool, the signature validation process, steps to update the firmware/software and error messages for failed situations.

2.38.3 KMD Evaluation Activity

None.

2.38.4 Test Evaluation Activity

The evaluators shall perform the following tests (if the TOE supports multiple signatures, each using a different hash algorithm, then the evaluator performs tests for different combinations of authentic and unauthentic digital signatures and hashes, as well as for digital signature alone): *Test 1: The evaluator performs the version verification activity to determine the current version of the TOE. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.* *Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that an update successfully installs on the TOE. The evaluator shall perform a subset of other evaluation activity tests to demonstrate that the update functions as expected.*

Test Number	1
Test Objective	Verify the TOE correctly reports its firmware version, rejects updates initiated by unauthorized users, rejects update whose signature is corrupted, signature key is untrusted, and installs update with a valid signature.
Test Steps Performed	This test was performed on Model A and Model B. The Evaluator initialized the TOE, started a SATA data capture, and checked the current executing version of the TOE firmware. The Evaluator then attempted to perform a firmware update from the ‘Login State’ and

	verified that the attempt failed. The evaluator then authenticated to the TOE, attempted to update the TOE firmware using a corrupted firmware update image, and verified that the attempt was unsuccessful. The evaluator then authenticated to the TOE, attempted to update the TOE firmware using a firmware update image that was signed with an untrusted key, and verified that the attempt was unsuccessful. The evaluator then authenticated to the TOE, attempted to update the TOE firmware using a good firmware update image, and verified that the attempt was successful. The Evaluator then power-cycled the TOE, authenticated to the TOE, queried the TOE's current executing firmware and verified that the firmware version was updated as expected.
Test Result	Pass

Test Number	2
Test Objective	Verify that successfully installed firmware update functions as expected.
Test Steps Performed	This test was performed on Model A and Model B. Continuing from Test 1 above, the Evaluator authenticated to the TOE and verified that User data was accessible. The Evaluator then changed the password to a new password, attempted to authenticate using the old password and verified that the attempt failed. The Evaluator attempted to authenticate using the new good password and verified that the attempt was successful. The Evaluator then logged out of the TOE, attempted to authenticate to the TOE 11 times using an incorrect password and verified that the TOE performed a zeroization of the user data after 11 failed authentication attempts.
Test Result	Pass

3 SAR Assurance Activities and Results

3.1 ASE: Security Target Evaluation

3.1.1 Conformance Claims (ASE_CCL)

3.1.2 ASE_CCL.1-8C

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

The evaluator determined that the Security Problem Definition in the [ST] is identical to Security Objectives in [PP1] and [PP2].

3.1.3 ASE_CCL.1-9C

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

The evaluator determined that the Security Objectives in the [ST] are identical to Security Objectives in [PP1] and [PP2].

3.1.4 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.

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

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

The evaluator verified that all SFRs in the Statement of Security Requirements in the [ST] are mandatory SFRs, optional SFRs, or appropriate selection-based SFRs as defined by the [PP1] and [PP2].

3.2 ADV: Development

3.2.1 Basic Functional Specification (ADV_FSP.1)

3.2.2 ADV_FSP.1-1

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

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.

[ST] Section 7.2.1 identifies 2 interfaces for the TOE: SATA or PCIe (M.2), and GPIO.

[AGD] Section 4 describes external TOE interfaces for different models of the TOE, SATA, PCIE and GPIO control interface when present.

[AGD] Section 19 describes that the TOE uses a modified SATA or M.2 pin layout. [AGD] Section 10 describes how the model number identifies those pin layout modifications.

[AGD] Section 3 describes that the TOE evaluated functionality is configured by using standard ATA/NVM commands.

[ST] Section 7.2.1 identifies that a SATA/PCIe interface is used for control functionality and data input\output, while a GPIO interface is used to enforce Key Destruction TSF.

[SCG] contains a list of supported commands and a list of methods for using the ATA and PCIe (NVM) interface.

[AGD] Section 7 describes that GPIO is used by connecting SE pin to GND pin.

3.2.3 ADV_FSP.1-2

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.

[SCG] contains a list of supported commands and a list of methods for using the ATA and PCIe (NVM) interface.

[AGD] Section 7 describes that GPIO is used by connecting SE pin to GND pin.

3.2.4 ADV_FSP.1-3

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.

[SCG] contains a list of supported commands and a list of methods for using the ATA and PCIe (NVM) interface. This includes supported parameters.

3.2.5 ADV_FSP.1-4

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.

The developer-provided documentation enables identification of all SFR-enforcing and SFR-supporting interfaces of the TOE; therefore, this Work Unit is satisfied.

3.2.6 ADV_FSP.1-5

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), 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'.

The Evaluator used [ST] and [AGD] to develop a mapping of the interfaces to SFRs.

SFRs			Interface	
#	SFR	Description	SATA/PCIe	GPIO
1	FCS_AFA_EXT.1	Authorization Factor Acquisition		
2	FCS_AFA_EXT.2	Timing of Authorization Factor Acquisition		
3	FCS_CKM.1(b)	Cryptographic key generation (Symmetric Keys) (Selection-based)		
4	FCS_CKM.1(c)	Cryptographic key generation (Data Encryption Key)		
5	FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)		
6	FCS_CKM.4(b)	Cryptographic Key Destruction (TOE-Controlled Hardware) (Selection-based)		X
7	FCS_CKM.4(d)	Cryptographic Key Destruction (Software TOE, 3rd Party Storage)		
8	FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)		
9	FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)		
10	FCS_CKM_EXT.6	Cryptographic Key Destruction Types		X
11	FCS_COP.1(a)	Cryptographic Operation (Signature Verification) (Selection-based)		
12	FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm) (Selection-based)		
13	FCS_COP.1(c)	Cryptographic Operation (Message Authentication) (Selection-based)		
14	FCS_COP.1(d)	Cryptographic operation (Key Wrapping) (Selection-based)		
15	FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption) (Selection-based)	X	
16	FCS_KDF_EXT.1	Cryptographic Key Derivation (Selection-based)	X	
17	FCS_KYC_EXT.1	Key Chaining (Initiator)	X	
18	FCS_KYC_EXT.2	Key Chaining (Recipient)		
19	FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning (Selection-based)	X	
20	FCS_RBG_EXT.1	Random Bit Generation (Selection-based)		
21	FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)		
22	FCS_VAL_EXT.1	Validation	X	
23	FDP_DSK_EXT.1	Protection of Data on Disk	X	
24	FMT_MOF.1	Management of Functions Behavior	X	

SFRs			Interface	
#	SFR	Description	SATA/PCIe	GPIO
25	FMT_SMF.1	Specification of Management Functions	X	X
26	FMT_SMR.1	Security Roles	X	
27	FPT_FUA_EXT.1	Firmware Update Authentication (Selection-based)	X	
28	FPT_KYP_EXT.1	Protection of Key and Key Material	X	
29	FPT_PWR_EXT.1	Power Saving States		
30	FPT_PWR_EXT.2	Timing of Power Saving States		
31	FPT_RBP_EXT.1	Rollback Protection (Optional)	X	
32	FPT_TST_EXT.1	TSF Testing	X	
33	FPT_TUD_EXT.1	Trusted Update	X	

3.2.7 ADV_FSP.1-6

EAs that are associated with the SFRs in [SD1], [SD2] 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.

This Work Unit is satisfied by corresponding evaluation activities in Section 2.

3.2.8 ADV_FSP.1-7

EAs that are associated with the SFRs in [SD1], [SD2] 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.

This Work Unit is satisfied by corresponding evaluation activities in Section 2.

3.3 AGD: Guidance Documents

3.3.1 Operational User Guidance (AGD_OPE.1)

3.3.2 AGD_OPE.1-1

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.

[ST] Section 1.4 contains a reference to the [AGD] and to the [SCG], identifying them as a part of the TOE, which ensures that users and administrators are aware of the preparatory procedures.

[AGD] Section 13 and Table 18 contain references to the [SCG] and describe that evaluation was performed using commands described in this document.

3.3.3 AGD_OPE.1-2

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.

[ST] Section 1.4.1 describes the physical scope of the TOE. This section also identifies the models that the documentation applies to. [AGD] Section 3 lists the models with specified hardware, firmware and operational environment requirements.

TOE developer Original Part No.	HW Ver.	Description (Form factor & Interface)	Firmware Ver.	User Capacity	Certification Sponsor Reseller Part No.
NS361P500GCCR-1F	04MB3	2.5" SATA 7mm MLC 500GB	NV.R1900_1000	500GB	AMP25T500-IM02AI
NS371P02T0CC1-1F	08MN3	2.5" SATA 7mm MLC 2TB	NV.R1900_1000	2TB	AMP25TT20-IM02AI
NS371P04T0CC1-1F	16MN3	2.5" SATA 7mm MLC 4TB	NV.R1900_1000	4TB	AMP25TT40-IM02AI
NS371P10T0CC0-1F	16MN3	2.5" SATA 9.5mm MLC 10TB	NV.R1900_1000	10TB	AMP25TT10-IM02AI
NS561P500GCE7-1F	02MB3	M.2 2280 PCIe/NVMe MLC 500GB	NV.R1900_1000	500GB	AMPW5D500-IM02AI
NS571P02T0CK7-1F	16SN3	M.2 22110 PCIe/NVMe MLC 2TB	NV.R1900_1000	2TB	AMPW6DT20-IM02AI
NS571P08T0CC0-1F	16MN3	2.5” PCIe/NVMe (U.2) MLC 8TB	NV.R1900_1000	8TB	AMP2UDT80-IM02AI

3.3.4 AGD_OPE.1-3

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.

[AGD] Section 2 states that the TOE was not evaluated or tested with other cryptographic engines.

[SCG] Section 4 contains the secure configuration procedure through ATA commands sequence.

[AGD] Section 13, Section 9 and Section 30 contain guidance for the Security Administrator regarding configuring the TOE for use. This configuration does not include configuration of actual cryptographic engine which is not configurable in the TOE.

3.3.5 AGD_OPE.1-4

The operational guidance shall describe how to configure the IT environments that are supported to shut down after an administratively defined period of inactivity.

[AGD] Section 14 states “the CO and system designers must implement host system application techniques, safeguards, and/or procedures that remove power from the TOE whenever the host platform is left unattended. Upon removal of power, the TOE purges the DEK and moves to a complete power-off state in less than 20 milliseconds.”

3.3.6 AGD_OPE.1-5

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.

[AGD] Section 6 describes the TOE enforces operation in power states D0 (fully on) and D3 (cold).

[AGD] Section 6 states that the TOE does not support any other power saving states than D0 and D3, so it does not support “sleeping” state.

[AGD] Section 6 clearly describes how the TOE state changes when the platform operating system changes power saving states.

3.3.7 AGD_OPE.1-6

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.

[AGD] Section 11 describes that the TOE supports CC compliant modes. The functionality coved by the evaluation activities is the mode with TOE Security Mode Enabled. No other key management modes were evaluated during the course of the CC-Evaluation.

[AGD] Section 7 describes Military Secure Erase protocols and states that this functionality is not covered by evaluation activities.

[AGD] Section 24 describes the Write Protect function of the TOE and contains a statement that this function is not evaluated during evaluation.

3.3.8 Preparative Procedures (AGD_PRE.1)

3.3.9 AGD_PRE.1-1

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.

[ST] Section 1.4 contains the reference to the [AGD] and to the [SCG] as a TOE part, that ensures that users and administrators are aware of the preparatory procedures.

[AGD] Section 13 and Table 18 contain references to the [SCG].

3.3.10 AGD_PRE.1-2

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).

[AGD] Section 12 states, “Scalar P-series SSD will function correctly in all host system that include a standard SATA interface and are compliant to the SATA and ATA8 specification.”

[AGD] Section 12 states, “Express P-series SSD will function correctly in all host system that include a standard PCIe interface and are compliant to the PCIe and NVMe specification.”

[AGD] Section 13 lists the assumptions about the operational environment. The evaluator confirmed that these assumptions are consistent with the OE requirements listed in [ST] Sections 1.3.4 and 4.1.

Security Objective	Rationale
OE.TRUSTED_CHANNEL	The boundary between the AA and the EE is within the TOE boundary; therefore, this objective is implicitly satisfied.
OE.INITIAL_DRIVE_STATE	[AGD] Section 13 states, “The administrator or CO understands that Novachips supplies the TOE in Security Mode Disabled (or Uninitialized State). The TOE contains no data when delivered by Novachips. The administrator or CO shall not store information on the TOE until after completing the initial security configuration procedure.”
OE.PASSPHRASE_STRENGTH	[AGD] Section 13 states, “The TOE accepts passwords length

	minimum 10 bytes up to 64 bytes. The administrator shall enforce complexity to provide suitable security strength. “
OE.POWER_DOWN[4]	[AGD] Section 13 states, “The administrator and system designer shall implement application techniques, safeguards, and/or procedures to assure that power is removed from the TOE, state D3 (cold), when the host system is left unattended. On removal of power, the TOE purges the DEK and enters a full-off state in less than 20 miliseconds.”
OE.POWER_DOWN[5]	[AGD] Section 13 states, “The administrator and system designer shall implement application techniques, safeguards, and/or procedures to assure that power is removed from the TOE, state D3 (cold), when the host system is left unattended. On removal of power, the TOE purges the DEK and enters a full-off state in less than 20 miliseconds.”
OE.SINGLE_USE_ET	The TOE does not support authorization tokens; this objective is implicitly satisfied.
OE.TRAINED_USERS	[AGD] Section 13 states, “The administrator CO shall train any users involved in the provisioning of the TOE in the methods and procedures to properly handle, store, and secure the Host Key values. For example, the Host Key value should be stored separately from the TOE”
OE.STRONG_ENVIRONMENT_CRYPTO	[AGD] Section 13 states, “The TOE does not support TCG Opal, or require a trusted platform module for secure operation.” [ST] Section 7.1.3 describes that TOE itself implements all cryptographic functions.
OE.PLATFORM_STATE	[AGD] Section 13 states, “The administrator or CO shall implement methods and procedures to assure that the host system is free of malware that could interfere with the correct operation and power-off procedures of the host system connected to the TOE.”
OE.PLATFORM_I&A	[AGD] Section 13 states, “The TOE does not interfere with or change the normal platform identification and authentication functionality such as the operating system login.”
OE.PHYSICAL	[AGD] Section 13 states, “The TOE is located in secure environment during initial secure configuration.”

[AGD] Section 13 describes the steps security administrators shall perform to ensure the operational environment is suitable for secure operation of the TOE.

3.3.11 AGD_PRE.1-3

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.

[ST] Section 1.4.1 describes the physical scope of the TOE. This section also identifies the TOE models. [AGD] Section 3 lists all the TOE models and specifies the hardware, firmware, and formfactor/interface type.

The models of hard drives in the TOE require either 2.5” SATA, NVMe, or M.2 SATA connections in the OE. [AGD] Section 4 identifies the required interface and connector type and associates the requirement on a per-model basis.

[4] As defined in the AA Protection Profile
[5] As defined in the EE Protection Profile

3.3.12 AGD_PRE.1-4

The preparative procedures must include

a) instructions to successfully install the TSF in each Operational Environment; and

b) instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and

c) instructions to provide a protected administrative capability.

For a):

[ST] Section 1.4.1 describes the physical scope of the TOE. This section also identifies the models that the documentation applies to. The models of hard drives in the TOE require either 2.5" SATA, NVMe, or M.2 SATA connections in the OE. [AGD] identifies the required connector type and associates the requirement on a per-model basis.

[AGD] Section 19 describes the installation of the TOE into a host system. Section 3 lists the hardware models of TOE. This list is consistent with the models listed in the [ST]. Section 4 describes ports and logical interfaces.

Section 16 describes the initial secure configuration of the TOE can be accomplished by using admin tool. This section describes steps to launch the admin tool to begin configuration.

Section 17 describes the Command Interface. The TOE supports two logical command interfaces, the guidance of this is in [SCG] Sections 5 and 6.

For b):

[AGD] Section 30 describes the steps security administrators have to perform to install the device. [AGD] Section 26 describes the physical security mechanisms of the TOE. [AGD] Section 13 contains assumptions and requirements about the operating environment. [AGD] Section 20 describes the critical security parameters, public keys and private keys.

[SCG] Section 4 contains the secure configuration procedure through ATA commands sequence or using NVM vendor-defined command set.

The evaluator determines that these are sufficient instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

For c):

[AGD] Sections 15 and 30 contain instructions for providing administrative capability.

3.4 ALC: Life-cycle Support

3.4.1 Labelling of the TOE (ALC_CMC.1)

3.4.2 ALC_CMC.1-1

The evaluator shall check that the TOE provided for evaluation is labelled with its reference.

The evaluator should ensure that the TOE contains the unique reference which is stated in the ST. This could be achieved through labelled packaging or media, or by a label displayed by the operational TOE. This is to ensure that it would be possible for consumers to identify the TOE (e.g. at the point of purchase or use).

The TOE may provide a method by which it can be easily identified. For example, a software TOE may display its name and version number during the start up routine, or in response to a command line entry. A hardware or firmware TOE may be identified by a part number physically stamped on the TOE.

Alternatively, the unique reference provided for the TOE may be the combination of the unique reference of each component from which the TOE is comprised (e.g. in the case of a composed TOE).

For Model A:

The evaluator examined the TOE Model A and verified that it has labeling containing a model identification (see Figure 1).

For Model B:

The evaluator examined the TOE Model B and verified that it has labeling containing a model identification (see Figure 3).

3.4.3 ALC_CMC.1-2

The evaluator shall check that the TOE references used are consistent.

If the TOE is labelled more than once then the labels have to be consistent. For example, it should be possible to relate any labelled guidance documentation supplied as part of the TOE to the evaluated operational TOE. This ensures that consumers can be confident that they have purchased the evaluated version of the TOE, that they have installed this version, and that they have the correct version of the guidance to operate the TOE in accordance with its ST.

The evaluator also verifies that the TOE reference is consistent with the ST.

If this work unit is applied to a composed TOE, the following will apply. The composed IT TOE will not be labelled with its unique (composite) reference, but only the individual components will be labelled with their appropriate TOE reference. It would require further development for the IT TOE to be labelled, i.e. during start-up and/or operation, with the composite reference. If the composed TOE is delivered as the constituent component TOEs, then the TOE items delivered will not contain the composite reference. However, the composed TOE ST will include the unique reference for the composed TOE and will identify the components comprising the composed TOE through which the consumers will be able to determine whether they have the appropriate items.

For Model A:

The evaluator examined the TOE Model A and verified that it has labeling consistent with the model identification presented in the [ST].

For Model B:

The evaluator examined the TOE Model A and verified that it has labeling consistent with the model identification presented in the [ST].

3.4.4 TOE CM Coverage (ALC_CMS.1)

3.4.5 ALC_CMS.1-1

The evaluator shall check that the configuration list includes the following set of items:

a) the TOE itself;

b) the evaluation evidence required by the SARs in the ST.

[CMS] contains a configuration list that includes TOE model identification and firmware version.

[CMS] contains a list of evidence provided for evaluation: Security Target, Guidance, and Entropy Analysis Report.

3.4.6 ALC_CMS.1-2

The evaluator shall examine the configuration list to determine that it uniquely identifies each configuration item.

The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

[CMS] contains identification for TOE hardware version, firmware version, and Git ID for version control.

[CMS] contains the date and version for the Security Target, Guidance documents, and Entropy Analysis Report.

3.5 ATE: Independent Testing (ATE_IND.1)

3.5.1 ATE_IND.1

The evaluator shall examine the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST. The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state. The evaluator shall prepare a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities. While it is not necessary to have one test case per test listed in an Evaluation Activity, the evaluator must show in the test plan that each applicable testing requirement in the SFR-related Evaluation Activities is covered. The test plan identifies the platforms to be tested, and for any platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition and configuration of each platform to be tested, and any setup actions that are necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of any cryptographic engine to be used (e.g. for cryptographic protocols being evaluated). The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives, and the expected results. The test report (which could just be an updated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure, so that a fix was then installed and then a successful re-run of the test was carried out, then the report would show a "fail" result followed by a "pass" result (and the supporting details), and not just the "pass" result.

[ST] Section 1.3.4 describes that the TOE requires 2.5'' SATA-compatible interface or M.2 compatible interface to operate. Testing was performed using 2.5" SATA interface for NS371P10T0CC0-1F model and M.2 interface for NS561P500GCE7-1F model.

The evaluator examined each model of the TOE before performing the evaluation activities. Each model has the same firmware model and control hash sum and was in uninitialized state.

This test report comprises the test documentation.

The evaluator conducted the testing.

This test report contains the required information for tests, composing the test subset. It contains instructions to set up test environment, the TOE condition at the beginning of each test, instructions to perform testing steps and expected results for each of the test steps. It also contains actual test results.

The evaluator checked that all actual test results are consistent with the expected test results.

The evaluator prepared the Evaluation Technical Report with the evaluation testing approach, configuration, methods, steps, and outcomes.

3.6 AVA: Vulnerability Assessment

3.6.1 Evaluation Activity (Documentation):

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components apply to all systems claimed in the ST, and should identify at a minimum the processors used by the TOE. Software components include any libraries used by the TOE, such as cryptographic libraries. This additional documentation is merely a list of the name and version number of the components, and will be used by the evaluators in formulating hypotheses during their analysis.
The evaluator shall examine the documentation outlined below provided by the vendor to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.
In addition to the activities specified by the CEM in accordance with Table 2 above, the evaluator shall perform the following activities.

[ST] Section 1.4.1 lists hardware components included in the TOE – Novachips NVS3800 ASIC SSD Controller, memory chips.

3.6.2 AVA_VAN Reporting

The evaluators shall produce two reports on the testing effort; one that is public-facing (that is, included in the non-proprietary evaluation report, which is a subset of the Evaluation Technical Report (ETR)) and the complete ETR that is delivered to the overseeing CB.
The public-facing report contains:
- *The flaw identifiers returned when the procedures for searching public sources were followed according to instructions in [SD] Section A.1.1;*
- *A statement that the evaluators have examined the Type 1 flaw hypotheses specified [SD] in section A.1.1 (i.e. the flaws listed in the previous bullet) and the Type 2 flaw hypotheses specified in the [SD] by the iTC in Section A.1.2;*

No other information is provided in the public-facing report.
The internal CB report contains, in addition to the information in the public-facing report:
- *a list of all of the flaw hypotheses generated (cf. AVA_VAN.1-4);*
- *the evaluator penetration testing effort, outlining the testing approach, configuration, depth and results (cf. AVA_VAN.1-9);*
- *all documentation used to generate the flaw hypotheses (in identifying the documentation used in coming up with the flaw hypotheses, the evaluation team must characterize the documentation so that a reader can determine whether it is strictly required by this Supporting Document, and the nature of the documentation (design information, developer engineering notebooks, etc.));*
- *the evaluator shall report all exploitable vulnerabilities and residual vulnerabilities, detailing for each:*
  - *its source (e.g. CEM activity being undertaken when it was conceived, known to the evaluator, read in a publication);*
  - *the SFR(s) not met;*
  - *a description;*
  - *whether it is exploitable in its operational environment or not (i.e. exploitable or residual).*
  - *the amount of time, level of expertise, level of knowledge of the TOE, level of opportunity and the equipment required to perform the identified vulnerabilities (cf. AVA_VAN.1-11);*

o how each flaw hypothesis was resolved (this includes whether the original flaw hypothesis was confirmed or disproved, and any analysis relating to whether a residual vulnerability is exploitable by an attacker with Basic Attack Potential) (cf. AVA_VAN1-10); and
o in the case that actual testing was performed in the investigation (either as part of flaw hypothesis generation using tools specified by the iTC in [SD] Section A.1.4, or in proving/disproving a particular flaw) the steps followed in setting up the TOE (and any required test equipment); executing the test; post-test procedures; and the actual results (to a level of detail that allow repetition of the test, including the following:
□ identification of the potential vulnerability the TOE is being tested for;
□ instructions to connect and setup all required test equipment as required to conduct the penetration test;
□ instructions to establish all penetration test prerequisite initial conditions;
□ instructions to stimulate the TSF;
□ instructions for observing the behaviour of the TSF;
□ descriptions of all expected results and the necessary analysis to be performed on the observed behaviour for comparison against expected results;
□ instructions to conclude the test and establish the necessary post-test state for the TOE. (cf. AVA_VAN.1-6, AVA_VAN.1-8).

Public-facing report.

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

After the Evaluator's review of all search hits identified above, 34 CVE records, relevant to the evaluated technology and TOE type, including one 'Vulnerability Note', remained and are listed below:

- CVE-2019-10636
- CVE-2019-10637
- CVE-2019-13466
- CVE-2019-13467
- CVE-2019-1589
- CVE-2019-6481
- CVE-2020-0407
- CVE-2020-11932
- CVE-2020-11933
- CVE-2020-12309
- CVE-2020-12310
- CVE-2020-12311
- CVE-2020-13799
- CVE-2020-26200
- CVE-2020-29063
- CVE-2020-3960
- CVE-2020-8701
- CVE-2020-8759
- CVE-2021-0100
- CVE-2021-0148
- CVE-2021-21522
- CVE-2021-23893
- CVE-2021-28653

- CVE-2021-33069
- CVE-2021-33074
- CVE-2021-33075
- CVE-2021-33077
- CVE-2021-33078
- CVE-2021-33080
- CVE-2021-33082
- CVE-2021-33083
- CVE-2021-3947
- CVE-2021-4210
- CVE-2022-25154
- Vulnerability Note VU#395981

The evaluator examined the Type 1 flaw hypotheses specified by [SD] in section A.1.1 (i.e., the flaws listed above) and the Type 2 flaw hypotheses specified in the [SD] by the iTC in Section A.1.2 and determined that no residual vulnerabilities exist in the TOE.

4 References

Abbr.	Name	Version	Date
[ST]	Scalar and Express P-series SSD, version NV.R1900 Security Target	1.0	June 6, 2022
[AGD]	Scalar and Express P-series SSD Non-Proprietary Administrative Guidance	1.0	March 3, 2022
[SCG]	Scalar and Express P-series SSD Non-Proprietary ATA/NVM Command Guidance	1.0	March 3, 2022
[CMS]	Configuration Management Specification for CC/CSfC	1.0	March 3, 2022