Scalar and Express P-series SSD, version NV.R1900 Security Target

UL13480549-ST

Version: 1.1

February 22, 2023

Prepared For:

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

Prepared By:

UL Verification Services Inc.

Table of Contents

1. Security Target Introduction

This Security Target (ST) is the statement of security needs for the specified Target of Evaluation (TOE). The structure of this document is defined by CC v3.1r5 Part 1 Annex A.2, "Mandatory contents of an ST":

- Section 1 contains the ST Introduction, including the ST reference, TOE reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions

1.1 Security Target Reference

The Security Target reference shall uniquely identify the Security Target.

ST Title: Scalar and Express P-series SSD, version NV.R1900 Security Target

ST Version Number: Version 1.1

ST Author(s): Dylan Lyman, Devin Becker, Gerrit Kruitbosch, UL Verification Services Inc.

ST Publication Date: February 22, 2023

Keywords Full Drive Encryption, Encryption Engine, Authorization Acquisition

1.2 Target of Evaluation Reference

The Target of Evaluation reference shall identify the Target of Evaluation.

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

TOE Name: Scalar and Express P-series SSD, version NV.R1900

TOE Version

The specific part numbers and HW and FW versions are shown in the following table:

Table 1: TOE Models

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	NV.R1900_1000	500GB	AMP25T500-IM02AI
NS371P01T0CC1-1F	04MN3	2.5" SATA 7mm	NV.R1900_1000	1TB	AMP2500T0T10-IM020CP
NS371P02T0CC1-1F	08MN3	2.5" SATA 7mm	NV.R1900_1000	2TB	AMP25TT20-IM02AI
NS371P04T0CC1-1F	16MN3	2.5" SATA 7mm	NV.R1900_1000	4TB	AMP25TT40-IM02AI
NS371P08T0CC0-1F	16MN3	2.5" SATA 9.5mm	NV.R1900_1000	8TB	AMP2500T08T0-IM020CP
NS371P10T0CC0-1F	16MN3	2.5" SATA 9.5mm	NV.R1900_1000	10TB	AMP25TT10-IM02AI
NS379P16T0VC0-1F	32MN1	2.5" SATA 9.5mm	NV.R1900_1000	16TB	AMP2500T16T0-IM020CP
NS379P20T0VC0-1F	32MN1	2.5" SATA 9.5mm	NV.R1900_1000	20TB	AMP2500T20T0-IM020CP
NS361P125GCM7-1F	04MBB	M.2 2242, SATA	NV.R1900_1000	125GB	AMPW300T0125-IM020CP
NS369P250GVM7-1F	04MBA	M.2 2242, SATA	NV.R1900_1000	250GB	AMPW300T0250-IM020CP
NS369P500GVM7-1F	04MBA	M.2 2242, SATA	NV.R1900_1000	500GB	AMPW300T0500-IM020CP
NS369P01T0VE7-1F	04MB1	M.2 2280, SATA	NV.R1900_1000	1TB	AMPW500T0T10-IM020CP
NS369P01T0VA7-1F	04MB1	mSATA SATA	NV.R1900_1000	1TB	AMPV500T0T10-IM020CP
NS569P500GVM7-1F	04MBA	M.2 2242, PCIe/NVMe	NV.R1900_1000	500GB	AMPW300D0500-IM020CP
NS561P500GCE7-1F	02MB3	M.2 2280 PCIe/NVMe	NV.R1900_1000	500GB	AMPW5D500-IM02AI
NS571P02T0CK7-1F	16SN3	M.2 22110 PCIe/NVMe	NV.R1900_1000	2TB	AMPW6DT20-IM02AI
NS579P04T0VK7-1F	16SN1	M.2 22110, PCIe/NVMe	NV.R1900_1000	4TB	AMPW600D04T0-IM020CP
NS571P08T0CC0-1F	16MN3	2.5” PCIe/NVMe (U.2)	NV.R1900_1000	8TB	AMP2UDT80-IM02AI

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is the Scalar and Express P-series SSD, version NV.R1900. The self-encrypting solid state drives, each consists of a single ASIC controller, volatile DRAM memory chips and non-volatile NAND. The SSDs are compatible with industry standard form factors such as 2.5” SATA hard drive, mini-SATA (mSATA), M.2 SATA, or NVMe M.2 & U.2 SSD slot.

1.3.2 TOE Usage

The TOE is used to protect data at rest on a device that is lost or stolen while powered off. The TOE stores all host data in encrypted form, including MBR and partition table data, which also facilitates cryptographic erasure via sanitization of the encryption key.

The TOE operates in one of the following 3 states:

- Uninitialized state (Drive operation is normal but access is not authenticated.)
- Login-State (Unauthenticated state. Drive presents a blank read-only “shadow disk”)
- User-State (Authenticated state. Normal operation where data is protected at rest.)

The uninitialized state provides no protection of data at rest as access is unrestricted. This mode of operation is not evaluated.

The TOE also supports Military Secure Erase protocols. These protocols allow the administrator to clean and purge the user storage data as specified by the protocol automatically without host PC equipment or control software. All of the secure erase protocols listed in the Guidance document perform the zeroize operation first to destroy all keys and key materials prior to proceeding to the next steps. While performing the military secure erase, the module shuts down all logical interfaces except transmitting an output signal on the activity signal pin and will resume the erase process automatically even if the power supply is interrupted until full process completion. The key zeroize operation is evaluated under the FCS_CKM class of SFRs, however, overwriting of user data performed as part of the TOEs Military Secure Erase functionality is unevaluated. The drive also has a write protect feature which is unevaluated.

1.3.3 TOE Major Security Features Summary

- Cryptographic Support
- User Data Protection
- Security Management
- Protection of the TOE Security Functionality (TSF)

1.3.4 TOE IT environment hardware/software/firmware requirements

The TOE relies on a host that is capable of communicating via the provided hardware interface – SATA or PCI Express.

For Scalar SATA products, the physical embodiment conforms to the EIA SFF-8201 specification, the mSATA form factor specification (Serial ATA International Organization Serial ATA Revision 3.4), or the M.2 form factor specification (PCI Express M.2 Specification Revision 3.0, Version 1.2). The electrical and software interface is the Serial ATA revision 3.1 specification. As such, they can interface to any environment that is compatible with standard 2.5” SATA hard drives, mSATA and M.2 SATA SSDs, respectively.

For Express PCIe products, the physical embodiment confirms to the PCIe M.2 Electromechanical Spec Rev1.0 or the U.2 form factor specification (Enterprise SSD Form Factor Version 1.0a SFF8639). The electrical and software interface complies with PCIe Base Specification Revision 2.0 and NVMe Revision 1.1a. As such, they can interface to any environment that is compatible with standard M.2 SSDs and U.2 SSDs, respectively.

1.4 Target of Evaluation Description

1.4.1 Target of Evaluation Physical Boundaries

The physical boundary of the TOE is the physical drive enclosure. Each TOE contains a Novachips NVS3800 ASIC SSD Controller that supports either SATA/ATA or PCIe/NVMe. The TOE communicates with a host computing system through these interfaces. The TOE has two GPIO lines through which either a machine, represented by a Microcontroller Unit (MCU), or operator can use to write protect or zeroize the TOE data written by the host.

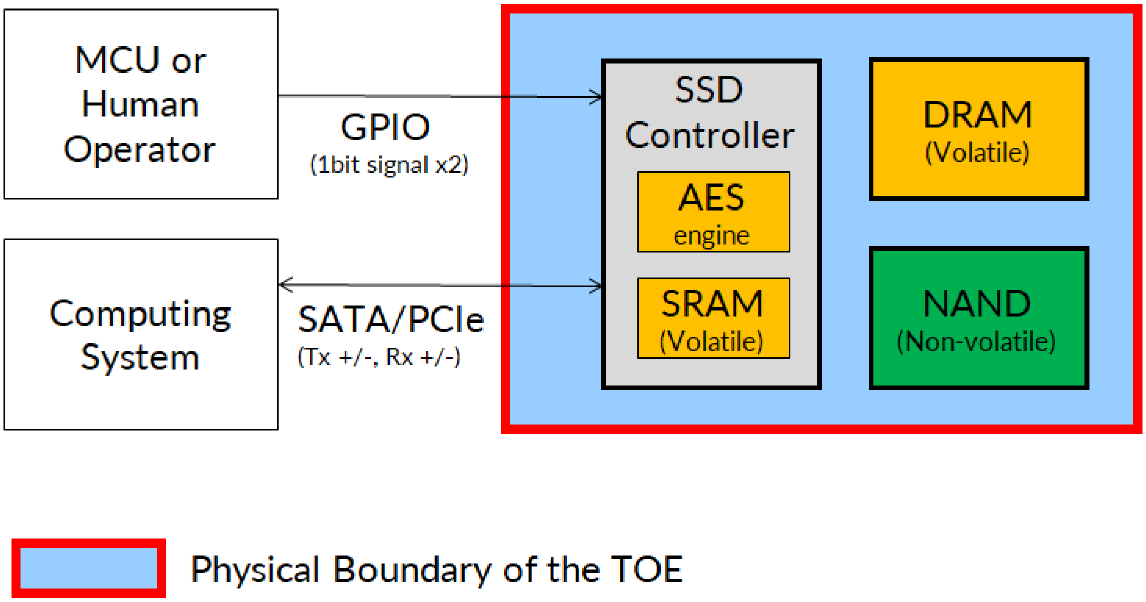

Figure 1: TOE Physical boundary

The specific part numbers that make up the various TOE configurations including the hardware version, firmware version and related properties is in Table 1 above. The TOE is delivered by the customer's trusted carrier or Novachips FEDEX/DHL account, and the tracking number is provided to the customer after arranging the shipment. The TOE is encapsulated by an aluminum enclosure with either tamper label or epoxy coating to detect any tamper evidence. Upon initialization, the admin or user should check for tamper evidence.

Novachips releases a firmware update as a single compressed zip file, which includes the firmware image and an update tool provided as an executable file for Windows or Linux OS environment. The customer can download the firmware update and other software from the Novachips support site after logging in with a unique username and password. This support site is managed by Novachips directly. Please contact your Novachips sales representative to have a support site login name and password generated.

The guidance documentation that is part of the TOE is listed in Section 9, "References," within Table 14: TOE Guidance Documentation. The guidance documentation is made available to consumers of the TOE by logging in to the vendor's customer support site and is offered as a download in a .pdf file format.

The TOE also supports the use of a software utility to aid in the use of managing and configuring the TOE. However, this functionality is not required for Common Criteria use and was not evaluated in this evaluation.

1.4.2 Target of Evaluation Logical Boundaries

The logical boundary of the TOE includes those security functions implemented exclusively by the TOE. These security functions are summarized in Section 1.3.3 above and are further described in the following subsections. A more detailed description of the implementation of each of these security functions are provided in Section 7, "TOE Summary Specification."

1.4.2.1 Cryptographic Support

The drive utilizes the following cryptographic algorithms that are approved for use by NIST FIPS 140-3 per SP 800-140C and SP 800-140D.

Table 2: Cryptographic Algorithms

Algorithm	Standard	Use	CAVP Cert. #
AES-KW	SP800-38F	Symmetric key wrapping	A897
AES-XTS-256	FIPS 197 SP800-38E	User data encryption and decryption	C448
DRBG	SP800-90A	Key, nonce and IV generation	C463
PBKDF	SP800-132	Key derivation using PBKDF option 2a	A897
SHA-256	FIPS 180-4	Used in DRBG and HMAC	C411
SHA-384	FIPS 180-4	Message Digest, Digital Signature	A897
HMAC-SHA-256	FIPS 198-1	Used in PBKDF	A897
ECDSA P-384	FIPS 186-4	Firmware image authentication using signature verification	A897

1.4.2.2 User Data Protection

The device uses XTS-AES-256 (SP800-38E) IEEE Std. 1619-2007 XTS-AES-256 algorithm to encrypt all user data on the drive.

1.4.2.3 Security Management

The TOE allows authorized users to change the data encryption key (DEK), erase the DEK, initiate firmware updates, erase user data, and change passwords.

1.4.2.4 Protection of the TSF

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.

1.5 Notation, Formatting, and Conventions

The notation, formatting, and conventions used in this Security Target are defined below; these styles and clarifying information conventions were developed to aid the reader.

The notation conventions that refer to iterations, assignments, selections, and refinements made in this Security Target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a Protection Profile.

The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; paragraph 6.4.1.3.2, “Permitted operations on components” as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

The notation used in the PP's to indicate iterations, assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document.

SFR component titles are annotated to indicate the source of the PP, e.g., (AA+PP) or (AA only). Iterations resulting from refinements or definitions that differ between the PPs are indicated by an identifier in parenthesis following each requirement functional element, e.g., FIA_UAU.1.1(EE).

Iterations performed in the Protection Profile are indicated by a letter in parenthesis following the requirement number, e.g., FCS_COP.1.1(c); the iterated requirement titles are similarly indicated, e.g., FCS_COP.1(c).

Assignments made by the ST author are identified with **bold text.**

Selections are identified with underlined text**.** Selections within selections are identified with double underlined text.

Refinements that add text are identified with ***bold and italicized text.*** Refinements that perform a deletion are identified with an underlined footnote reference.

2. Conformance Claims

2.1 Common Criteria Conformance Claims

This Security Target is conformant to the Common Criteria Version 3.1r5, CC Part 2 extended [3], and CC Part 3 conformant [4].

2.2 Conformance to Protection Profiles

This Security Target claims exact conformance to the collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0e, February 1, 2019 and the collaborative Protection Profile for Full Disk Encryption – Authorization Acquisition, Version 2.0e, February 1, 2019. These Protection Profiles will be referred to individually or collectively as FDE or cPP for convenience throughout this Security Target.

The TOE complies with the following Technical Decisions:

- 0464 – FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states
- 0460 – FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states
- 0458 – FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities

2.3 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements or security assurance requirements packages, neither as package-conformant or package-augmented.

2.4 Conformance Claims Rationale

In harmony with exact conformance, as described by CC and CEM addenda for Exact Conformance [11], the security problem definition, threats, organizational security policies, assumptions, security objectives, and security requirements are taken from the cPP. This ST does not alter or add to those defined in the cPP. Additionally, all SFRs and SARs defined in the cPP have been properly instantiated in this Security Target; therefore, this ST shows exact conformance to the cPP.

3. Security Problem Definition

3.1 Threats

The following table defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the cPP unchanged.

Table 3: Threats

Threat	Description
T.UNAUTHORIZED_DATA_ ACCESS	The cPP addresses the primary threat of unauthorized disclosure of protected data stored on a storage device. If an adversary obtains a lost or stolen storage device (e.g., a storage device contained in a laptop or a portable external storage device), they may attempt to connect a targeted storage device to a host of which they have complete control and have raw access to the storage device (e.g., to specified disk sectors, to specified blocks).
T.KEYING_MATERIAL_ COMPROMISE	Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. The cPP considers possession of keying material of equal importance to the data itself. Threat agents may look for keying material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash[, or TPMs][1].
T.AUTHORIZATION_GUESSING	Threat agents may exercise host software to repeatedly guess authorization factors, such as passwords and PINs. Successful guessing of the authorization factors may cause the TOE to release [BEV or DEKs][2] or otherwise put it in a state in which it discloses protected data to unauthorized users.
T.KEYSPACE_EXHAUST	Threat agents may perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms and/or parameters allow attackers to brute force exhaust the key space and give them unauthorized access to the data.
T.KNOWN_PLAINTEXT	Threat agents know plaintext in regions of storage devices, especially in uninitialized regions (all zeroes) as well as regions that contain well known software such as operating systems. A poor choice of encryption algorithms, encryption modes, and initialization vectors along with known plaintext could allow an attacker to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.
T.CHOSEN_PLAINTEXT	Threat agents may trick authorized users into storing chosen plaintext on the encrypted storage device in the form of an image, document, or some other file A poor choice of encryption algorithms, encryption modes, and initialization vectors along with the chosen plaintext could allow attackers to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.
T.UNAUTHORIZED_UPDATE	Threat agents may attempt to perform an update of the product which compromises the security features of the TOE. Poorly chosen update protocols, signature generation and verification algorithms, and parameters may allow attackers to install software [and/or firmware][3] that

---

[1] “or TPMS” not in the AA Protection Profile.

[2] “BEV” in the AA Protection Profile, “DEKs” in the EE Protection Profile.

[3] Not in the EE Protection Profile.

Table 3: Threats

Threat	Description
	bypasses the intended security features and provides them unauthorized to access to data.
T.UNAUTHORIZED_FIRMWARE_UPDATE	An attacker attempts to replace the firmware on the SED via a command from the AA or from the host platform with a malicious firmware update that may compromise the security features of the TOE.
T.UNAUTHORIZED_FIRMWARE_MODIFY	An attacker attempts to modify the firmware in the SED via a command from the AA or from the host platform that may compromise the security features of the TOE.

3.2 Organizational Security Policies

There are no organizational security policies addressed by the cPP or this ST.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the cPP unchanged.

Table 4: Assumptions

Assumption	Description
A.TRUSTED_CHANNEL	Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure. In cases in which a single product fulfils both cPPs, then the communication between the components does not extend beyond the boundary of the TOE (e.g., communication path is within the TOE boundary). In cases in which independent products satisfy the requirements of the AA and EE, the physically close proximity of the two products during their operation means that the threat agent has very little opportunity to interpose itself in the channel between the two without the user noticing and taking appropriate actions.
A. INITIAL_DRIVE_STATE	Users enable Full Drive Encryption on a newly provisioned [or initialized] [4] storage device free of protected data in areas not targeted for encryption. It is also assumed that data intended for protection should not be on the targeted storage media until after provisioning. The cPP does not intend to include requirements to find all the areas on storage devices that potentially contain protected data. In some cases, it may not be possible - for example, data contained in “bad” sectors. While inadvertent exposure to data contained in bad sectors or un-partitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors, unpartitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE preauthentication software) contain no protected data.

---

[4] Not in the EE Protection Profile

Table 4: Assumptions	
Assumption	**Description**
A.TRAINED_USER [5]	Authorized users follow all provided user guidance, including keeping password/passphrases and external tokens securely stored separately from the storage device and/or platform.
A.TRAINED_USER[6]	Users follow the provided guidance for securing the TOE and authorization factors. This includes conformance with authorization factor strength, using external token authentication factors for no other purpose and ensuring external token authorization factors are securely stored separately from the storage device and/or platform. The user should also be trained on how to power off their system.
A.PLATFORM_STATE	The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.
A.POWER_DOWN[7]	The user does not leave the platform and/or storage device unattended until all volatile memory is cleared after a power-off, so memory remnant attacks are infeasible. Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., lock screen). Users power the platform and/or storage device down or place it into a power managed state, such as a “hibernation mode”.
A.POWER_DOWN [8]	The user does not leave the platform and/or storage device unattended until the device is in a Compliant power saving state or has fully powered off. This properly clears memories and locks down the device. Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., lock screen or sleep state). Users power the platform and/or storage device down or place it into a power managed state, such as a “hibernation mode”.
A.STRONG_CRYPTO	All cryptography implemented in the Operational Environment and used by the product meets the requirements listed in the cPP. This includes generation of external token authorization factors by a RBG.
A.SECURE_STATE	Upon the completion of proper provisioning, the drive is only assumed secure when in a powered off state up until it is powered on and receives initial authorization.
A.SINGLE_USE_ET	External tokens that contain authorization factors are used for no other purpose than to store the external token authorization factors.
A.PASSWORD_STRENGTH	Authorized administrators ensure password/passphrase authorization factors have sufficient strength and entropy to reflect the sensitivity of the data being protected.
A.PLATFORM_I&A	The product does not interfere with or change the normal platform identification and authentication functionality such as the operating system login. It may provide authorization factors to the Operating system's login interface, but it will not change or degrade the functionality of the actual interface.
A.PHYSICAL	The platform is assumed to be physically protected in its Operational Environment and not subject to physical attacks that compromise the security and/or interfere with the platform’s correct operation.

---

[5] As defined in the AA Protection Profile
[6] As defined in the EE Protection Profile.
[7] As defined in the AA Protection Profile.
[8] As defined in the EE Protection Profile

4. Security Objectives

4.1 Security Objectives for the Operational Environment

Table 5: Security Objectives for the Operational Environment

Objective	Description
OE.TRUSTED_CHANNEL	Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure.
OE.INITIAL_DRIVE_STATE	The OE provides a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption.
OE.PASSPHRASE_STRENGTH	An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to guidance from the Enterprise using the TOE.
OE.POWER_DOWN[9]	Volatile memory is cleared after power-off so memory remnant attacks are infeasible.
OE.POWER_DOWN[10]	Volatile memory is cleared after entering a Compliant power saving state or turned off so memory remnant attacks are infeasible.
OE.SINGLE_USE_ET	External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.
OE.TRAINED_USERS	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.
OE.STRONG_ENVIRONMENT_CRYPTO	The Operating Environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE and Appendix A.
OE.PLATFORM_STATE	The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.
OE.PLATFORM_I&A	The Operational Environment will provide individual user identification and authentication mechanisms that operate independently of the authorization factors used by the TOE.
OE.PHYSICAL	The Operational Environment will provide a secure physical computing space such that an adversary is not able to make modifications to the environment or to the TOE itself.

---

[9] As defined in the AA Protection Profile

[10] As defined in the EE Protection Profile

5. Extended Components Definition

This section addresses the definition of the extended security functional and assurance requirements; the components that are CC Part 2 extended, and CC Part 3 extended, i.e., NIAP interpreted requirements, and extended requirements.

5.1 Extended Security Functional Requirements Definitions

In exact conformance to the Protection Profile(s) identified, this Security Target does not add to or modify the extended Security Functional Requirements defined by those Protection Profile(s). The Protection Profile(s) should be consulted for the content of the extended components definition.

5.2 Extended Security Assurance Requirements Definitions

There are no extended Security Assurance Requirements defined in this Security Target or the Protection Profile(s) it is conformant to.

6. Security Requirements

This section describes the security functional and assurance requirements for the TOE; those that are CC Part 2 conformant, CC Part 2 extended, CC Part 3 conformant.

6.1 Security Functional Requirements

This section describes the functional requirements for the TOE.

Table 6: Security Functional Requirements

#	SFR	Description
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)
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
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)
16	FCS_KDF_EXT.1	Cryptographic Key Derivation (Selection-based)
17	FCS_KYC_EXT.1	Key Chaining (Initiator)
18	FCS_KYC_EXT.2	Key Chaining (Recipient)
19	FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning (Selection-based)
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
23	FDP_DSK_EXT.1	Protection of Data on Disk
24	FMT_MOF.1	Management of Functions Behavior
25	FMT_SMF.1	Specification of Management Functions
26	FMT_SMR.1	Security Roles
27	FPT_FUA_EXT.1	Firmware Update Authentication (Selection-based)
28	FPT_KYP_EXT.1	Protection of Key and Key Material
29	FPT_PWR_EXT.1	Power Saving States

Table 6: Security Functional Requirements		
#	SFR	Description
30	FPT_PWR_EXT.2	Timing of Power Saving States
31	FPT_RBP_EXT.1	Rollback Protection (Optional)
32	FPT_TST_EXT.1	TSF Testing
33	FPT_TUD_EXT.1	Trusted Update

6.1.1 Class FCS: Cryptographic Support

6.1.1.1 FCS_AFA_EXT.1 (AA only) Authorization Factor Acquisition

FCS_AFA_EXT.1.1

The TSF shall accept the following authorization factors:

- <u>a submask derived from a password authorization factor conditioned as defined in FCS_PCC_EXT.1</u>.

6.1.1.2 FCS_AFA_EXT.2 (AA only) Timing of Authorization Factor Acquisition

FCS_AFA_EXT.2.1

The TSF shall reacquire the authorization factor(s) specified in FCS_AFA_EXT.1 upon transition from any Compliant power saving state specified in FPT_PWR_EXT.1 prior to permitting access to plaintext data.

6.1.1.3 FCS_CKM.1(b) (AA only) Cryptographic Key Generation (Symmetric Keys) (Selection-based)

FCS_CKM.1.1(b)

The TSF shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes <u>256 bit</u> that meet the following: No Standard.

6.1.1.4 FCS_CKM.1(c) (EE only) Cryptographic Key Generation (Data Encryption Key)

FCS_CKM.1.1(c)

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation method

- <u>generate a DEK using the RBG as specified in FCS_RBG_EXT.1</u>

and specified cryptographic key sizes <u>256 bits</u>.

6.1.1.5 FCS_CKM.4(a) (EE only) Cryptographic Key Destruction (Power Management)

FCS_CKM.4.1(a)

The TSF shall <u>erase</u> cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1 that meets the following: a key destruction method specified in FCS_CKM_EXT.6[11].

---

[11] FCS_CKM_EXT.6 in turn requires the selection of FCS_CKM.4(b), (c), and/or (d). The AA PP version of FCS_CKM.4.1(a) specifies FCS_CKM.4(d) specifically.

6.1.1.6 FCS_CKM.4(b) (EE only) Cryptographic Key Destruction (TOE-Controlled Hardware) (Selection-based)

FCS_CKM.4.1(b)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For volatile memory, the destruction shall be executed by a
    - single overwrite consisting of
        - zeroes,
        - **0x55**,
    - removal of power to the memory,
- For non-volatile memory
    - that does not employ a wear-leveling algorithm, the destruction shall be executed by a
        - block erase

      and if the read-verification of the overwritten data fails, the process shall be repeated again up to **0** times, whereupon an error is returned.

that meets the following: no standard.

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

FCS_CKM.4.1(d)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For volatile memory, the destruction shall be executed by a
    - single overwrite consisting of
        - zeroes,
        - 0x55,
    - removal of power to the memory;

that meets the following: no standard.

6.1.1.8 FCS_CKM_EXT.4(a) (AA+EE) Cryptographic Key and Key Material Destruction (Destruction Timing)

FCS_CKM_EXT.4.1(a)

The TSF shall destroy all keys and keying material when no longer needed.

6.1.1.9 FCS_CKM_EXT.4(b) (AA+EE) Cryptographic Key and Key Material Destruction (Power Management)

FCS_CKM_EXT.4.1(b)

The TSF shall destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1.

6.1.1.10 FCS_CKM_EXT.6 (EE only) Cryptographic Key Destruction Types

FCS_CKM_EXT.6.1

The TSF shall use FCS_CKM.4(b), FCS_CKM.4(d) key destruction methods.

6.1.1.11 FCS_COP.1(a) (AA+EE) Cryptographic Operations (Signature Verification) (Selection-based)

FCS_COP.1.1(a)

The TSF shall perform cryptographic signature services (verification) in accordance with a

- Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater

that meets the following:

- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-384; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes.

6.1.1.12 FCS_COP.1(b) (AA+EE) Cryptographic Operation (Hash Algorithm) (Selection-based)

FCS_COP.1.1(b)

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-256, SHA-384 that meet the following: ISO/IEC 10118-3:2004.

6.1.1.13 FCS_COP.1(c) Cryptographic Operation (Keyed Hash Algorithm - AA) (Message Authentication – EE) (Selection-based)

FCS_COP.1.1(c)(AA)

The TSF shall perform cryptographic keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-256 and cryptographic key sizes **256-bits** that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

FCS_COP.1.1(c)(EE)

The TSF shall perform cryptographic message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-256 and cryptographic key sizes **256-bits used in HMAC** that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

6.1.1.14 FCS_COP.1(d) (AA+EE) Cryptographic Operation (Key Wrapping) (Selection-based)

FCS_COP.1.1(d)

The TSF shall perform key wrapping in accordance with a specified cryptographic algorithm AES in the following modes KW and the cryptographic key size 256 bits that meet the following: AES as specified in ISO/IEC 18033-3 NIST SP 800-38F.

6.1.1.15 FCS_COP.1(f) (AA+EE) Cryptographic Operation (AES Data Encryption/Decryption) (Selection-based)

FCS_COP.1.1(f)

The TSF shall perform data encryption and decryption in accordance with a specified cryptographic algorithm AES used in XTS mode and cryptographic key sizes 256 bits that meet the following: AES as specified in ISO/IEC18033-3, and XTS as specified in IEEE 1619.

6.1.1.16 FCS_KDF_EXT.1 (AA+EE) Cryptographic Key Derivation (Selection-based)

FCS_KDF_EXT.1.1

The TSF shall accept a conditioned password submask, to derive an intermediate key, as defined in

- NIST SP 800-132

using the keyed-hash functions specified in FCS_COP.1(c), such that the output is at least of equivalent security strength (in number of bits) to the BEV.

6.1.1.17 FCS_KYC_EXT.1 (AA only) Key Chaining (Initiator)

FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of:

- intermediate keys originating from one or more submask(s) to the BEV using the following method
  - key derivation as specified in FCS_KDF_EXT.1

while maintaining an effective strength of 256 bits for symmetric keys and an effective strength of not applicable for asymmetric keys.

FCS_KYC_EXT.1.2

The TSF shall provide a 256 bit BEV to **the EE**

- after the TSF has successfully performed the validation process as specified in FCS_VAL_EXT.1.

6.1.1.18 FCS_KYC_EXT.2 (EE only) Key Chaining (Recipient)

FCS_KYC_EXT.2.1

The TSF shall accept a BEV of at least 256 bits from **the AA**.

FCS_KYC_EXT.2.2

The TSF shall maintain a chain of intermediary keys originating from the BEV to the DEK using the following method(s):

- key wrapping as specified in FCS_COP.1(d),

while maintaining an effective strength of 256 bits or symmetric keys and an effective strength of not applicable for asymmetric keys.

6.1.1.19 FCS_PCC_EXT.1 (AA only) Cryptographic Password Construct and Conditioning (Selection-based)

FCS_PCC_EXT.1.1

A password used by the TSF to generate a password authorization factor shall enable up to **64** characters in the set of {upper case characters, lower case characters, numbers, and **any other 8-bit value**} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-SHA-256 with **1000** iterations, and output cryptographic key sizes 256 bits that meet the following: NIST SP 800-132.

6.1.1.20 FCS_RBG_EXT.1 (AA+EE) Extended: Cryptographic Operation (Random Bit Generation) (Selection-based)

FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with NIST SP 800-90A using Hash_DRBG (any).

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from

- **One** hardware-based noise source(s)

with a minimum of 256 bits of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

6.1.1.21 FCS_SNI_EXT.1 (AA+EE) Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

FCS_SNI_EXT.1.1

The TSF shall use salts that are generated by a DRBG as specified in FCS_RBG_EXT.1.

FCS_SNI_EXT.1.2

The TSF shall use unique nonces with a minimum size of 64 bits.

FCS_SNI_EXT.1.3

The TSF shall create IVs in the following manner

- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer.

6.1.1.22 FCS_VAL_EXT.1 (AA+EE) Validation (Selection-based for AA)

FCS_VAL_EXT.1.1(AA)

The TSF shall perform validation of the

BEV using the following methods:

- Hash the BEV as specified in FCS_COP.1(b) and compare it to a stored hashed BEV.

FCS_VAL_EXT.1.1(EE)

The TSF shall perform validation of the BEV using the following method(s):

- key wrap as specified in FCS_COP.1(d).

FCS_VAL_EXT.1.2(AA)

The TSF shall require validation of the BEV prior to forwarding the BEV to the EE.

FCS_VAL_EXT.1.2(EE)

The TSF shall require the validation of the BEV prior to allowing access to TSF data after exiting a Compliant power saving state.

FCS_VAL_EXT.1.3

The TSF shall

- perform a key sanitization of the DEK upon a **10** of consecutive failed validation attempts.

6.1.2 Class FDP: User Data Protection

6.1.2.1 FDP_DSK_EXT.1 (EE only) Extended: Protection of Data on Disk

FDP_DSK_EXT.1.1

The TSF shall perform Full Drive Encryption in accordance with FCS_COP.1(f), such that the drive contains no plaintext protected data.

FDP_DSK_EXT.1.2

The TSF shall encrypt all protected data without user intervention.

6.1.3 Class FMT: Security Management

6.1.3.1 FMT_MOF.1 (AA only) Management of Functions Behavior

FMT_MOF.1.1

The TSF shall restrict the ability to modify the behaviour of the functions use of Compliant power saving state to authorized administrators.

6.1.3.2 FMT_SMF.1(AA + EE) Specification of Management Functions

FMT_SMF.1.1(AA)

The TSF shall be capable of performing the following management functions:

a) forwarding requests to change the DEK to the EE,
b) forwarding requests to cryptographically erase the DEK to the EE,
c) allowing authorized users to change authorization factors or set of authorization factors used,
d) initiate TOE firmware/software updates,
e) no other functions.

FMT_SMF.1.1(EE)

The TSF shall be capable of performing the following management functions:

a) Change the DEK, as specified in FCS_CKM.1, when re-provisioning or when commanded,
b) erase the DEK, as specified in FCS_CKM.4(a),
c) initiate TOE firmware/software updates,
d) zeroize user data.

6.1.3.3 FMT_SMR.1 (AA only) Security Roles

FMT_SMR.1.1

The TSF shall maintain the roles authorized user.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.1.4 Class FPT: Protection of the TSF

6.1.4.1 FPT_FUA_EXT.1 (EE only) Firmware Update Authentication (Selection-based)

FPT_FUA_EXT.1.1

The TSF shall authenticate the source of the firmware update using the digital signature algorithm specified in FCS_COP.1(a) using the RTU that contains hash value of the public key as specified in FCS_COP.1(b).

FPT_FUA_EXT.1.2

The TSF shall only allow installation of update if the digital signature has been successfully verified as specified in FCS_COP.1(a).

FPT_FUA_EXT.1.3

The TSF shall only allow modification of the existing firmware after the successful validation of the digital signature, using a mechanism as described in FPT_TUD_EXT.1.2.

FPT_FUA_EXT.1.4

The TSF shall return an error code if any part of the firmware update process fails.

6.1.4.2 FPT_KYP_EXT.1 (AA+EE) Extended: Protection of Key and Key Material [TD0458[12]]

FPT_KYP_EXT.1.1(AA)

The TSF shall

- only store plaintext keys that meet any one of the following criteria
    - The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1,
    - The plaintext key will no longer provide access to the encrypted data after initial provisioning.

FPT_KYP_EXT.1.1(EE)

The TSF shall

- only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1(d), or encrypted, as specified in FCS-COP.1(g) or FCS_COP.1(e)
- only store plaintext keys that meet any one of the following criteria
    - The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.2
    - The plaintext key will no longer provide access to the encrypted data after initial provisioning.

6.1.4.3 FPT_PWR_EXT.1 (AA+EE) Power Saving States [TD0460[13]][TD0464[14]]

FPT_PWR_EXT.1.1

The TSF shall define the following Compliant power saving states: **D3**[15].

---

[12] The evaluation activities were modified by TD0458.

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

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

[15] Assignment in AA. Selection in EE.

6.1.4.4 FPT_PWR_EXT.2 (AA+EE) Timing of Power Saving States

FPT_PWR_EXT.2.1

For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur: user-initiated request, shutdown, no other conditions.

6.1.4.5 FPT_RBP_EXT.1 (EE only) Rollback Protection (Optional)

FPT_RBP_EXT.1.1

The TSF shall verify that the new firmware package is not downgrading to a lower security version number by **checking the firmware header, which includes a version number**.

FPT_RBP_EXT.1.2

The TSF shall generate and return an error code if the attempted firmware update package is detected to be an invalid version.

6.1.4.6 FPT_TST_EXT.1 (AA+EE) Extended: TSF Testing[16]

FPT_TST_EXT.1.1(startup)

The TSF shall run a suite of the following self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF:

- **Firmware Image Verification**
- **SHA-256 KAT**
- **SHA-384 KAT**
- **HASH DRBG KAT**
- **HASH DRBG Health Tests**
- **AES-XTS Encrypt KAT**
- **AES-XTS Decrypt KAT**
- **NDRNG Repetition Count Test**
- **NDRNG Adaptive Proportion Test**
- **HMAC SHA-256**
- **PBKDF2**
- **Key Wrapping**
- **ECDSA P-384**.

FPT_TST_EXT.1.1(conditional)

The TSF shall run a suite of the following self-tests at the conditions before the function is first invoked to demonstrate the correct operation of the TSF:

- **HASH DRBG KAT**
- **HASH DRBG Health Tests**
- **NDRNG Repetition Count Test**
- **NDRNG Adaptive Proportion Test**

6.1.4.7 FPT_TUD_EXT.1 (AA+EE) Trusted Update

FPT_TUD_EXT.1.1

---

[16] Optional for AA

The TSF shall provide authorized users the ability to query the current version of the TOE firmware.

FPT_TUD_EXT.1.2

The TSF shall provide authorized users the ability to initiate updates to TOE firmware.

FPT_TUD_EXT.1.3(AA)

The TSF shall verify updates to the TOE software using a digital signature as specified in FCS_COP.1(a) by the manufacturer prior to installing those updates.

FPT_TUD_EXT.1.3(EE)

The TSF shall verify updates to the TOE firmware using a***n*** authenticated firmware update mechanism as described in FPT_FUA_EXT.1 by the manufacturer prior to installing those updates.

6.2 Security Assurance Requirements

The TOE security assurance requirements are taken from the cPPs with the refinements documented. They are identified in Table 7 below.

Table 7: Assurance Requirements

Assurance Class	Assurance Component
Security Target (ASE)	Conformance Claims (ASE_CCL.1)
	Extended Components Definition (ASE_ECD.1)
	ST Introduction (ASE_INT.1)
	Security Objectives for the Operational Environment (ASE_OBJ.1)
	Stated Security Requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE Summary Specification (ASE_TSS.1)
Development (ADV)	Basic Functional Specification (ADV_FSP.1)
Guidance documents (AGD)	Operational User Guidance (AGD_OPE.1)
	Preparative Procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM Coverage (ALC_CMS.1)
Tests (ATE)	Independent Testing – Sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability Survey (AVA_VAN.1)

The assurance elements are taken from the CEM [6] as modified by the CC and CEM addenda for exact conformance [11] and as refined by the cPPs as well as the following refinement.

ASE_TSS.1.1C

The TOE summary specification shall describe how the TOE meets each SFR, including a proprietary [17] Entropy Essay.

---

[17] "Key Management Description '(Appendix E)'" has been removed since it is included in this ST and not a separate or proprietary document.

7. TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following families of Security Functions:

- Cryptographic Support
- User Data Protection
- Security Management
- Protection of the TSF

7.1 Cryptographic Support

7.1.1 Authorization Factor

The TOE supports a single authorization factor of 10 to 64 bytes. While supporting ATA and NVM standards, the TOE can accept any 8-bit value specified by the user as the authorization factor.[18] These bytes may contain ASCII encoded values and special characters or any combination of ones and zeroes making up the “password”. Administrators must establish and enforce password content requirements to ensure suitable security strength. This authorization factor or “password” is directly input to the PBKDF function to produce the 256-bit Key Wrapping Key (KWK) that is the Border Encryption Value (BEV) passed from the Authorization Acquisition (AA) portion of the TOE to the Encryption Engine (EE) portion of the TOE.

When initially set, the TOE saves a SHA-384 hash of the authorization factor in order to validate the authorization factor upon subsequent authentication attempts. The TOE uses no other submasks, or combination of submasks, for user authentication. The password-based authorization factor is always required when resuming from the only compliant power-saving state, D3 (off), supported by the TOE. Failed validation attempts are tracked through a persistent counter that is reset after successful login or when zeroizing.

FCS_AFA_EXT.1.1, FCS_AFA_EXT.2.1, FCS_PCC_EXT.1.1, FCS_VAL_EXT.1

7.1.2 Cryptographic Key Management

The TOE keychain:

The TOE keychain is represented in Figure 2, and consists of the authorization factor or password and derived and protected keys including the data encryption key. The authorization factor is input to the PBKDF function along with a randomly generated 256 bit salt to produce the KWK or BEV. During provisioning the KWK is used to wrap a randomly generated DEK before being stored in a host inaccessible portion of NAND flash. Further understanding of the memory components, content, relationship and user accessibility during operation is in section 7.2.1. After provisioning and subsequent power on the module enters the Login-State or unauthenticated state. Once the authorization factor is supplied and checked against a known hash, it is used to generate the KWK which unwraps the DEK used to AES-XTS encrypt and decrypt the user data stream.

---

[18] Note that host systems may impose their own restrictions on authorization factors. See Command Guidance [2] for ATA support of authorization factors greater than 32 bytes.

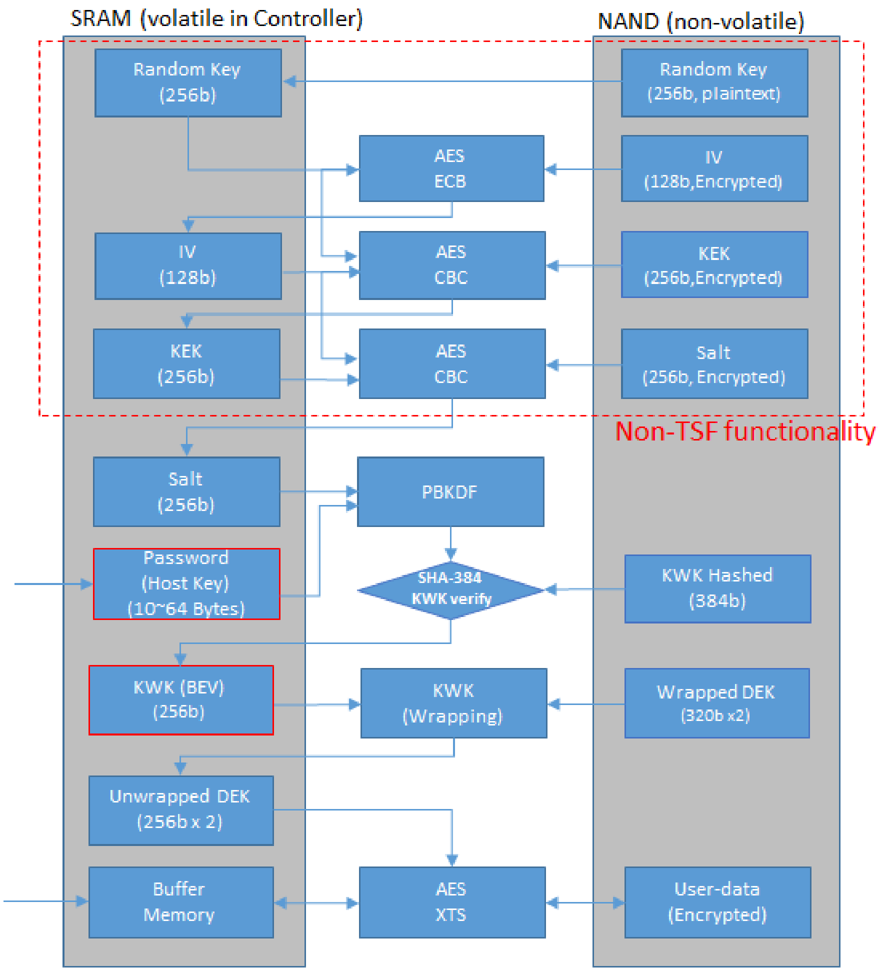

Figure 2: Cryptographic keychain

The TOE is a Self-Encrypting Drive and automatically generates, manages and destroys keys, as all keys are stored within the physical enclosure of the drive and not directly accessible by the end user through any interface. While the host can trigger zeroization or state changes, the TOE does not interact with the host to perform destruction. The TOE internal memory types are represented above in Figure 1 of section 1.4.1. The TOE includes DRAM volatile memory, used only for disk I/O operations as discussed in Section 7.2 below. No key information is ever present in DRAM. All volatile SRAM access is byte level random access and allocated as either embedded software process stack or memory pool, also not directly accessible to the user. The TOE's contains NAND flash for persistent storage and is erased at a per block unit. All persistent data used by the internal controller, including protected keys and non-keychain system information, is stored in a single block of NAND inaccessible to the host. Traditional background task-oriented wear-levelling of this host inaccessible NAND memory block is not performed. The remainder of NAND memory is for the storage of encrypted user data. Upon a change of password or zeroize the TOE immediately zeroizes the special NAND block and allocates a new block for storage. The KWK which functions as the BEV is never stored persistently and is therefore the only temporary key. The TOE does not persistently store plaintext keys that are part of the keychain. The TOE does not support manual key entry or any other type of key entry/output. The keys in the key chain are listed in Table 8.

Table 8: Keychain

Security Parameter / Key	Length (Bits)	Initialization	Usage	Persistent Storage
Authorization factor	80 – 512	User-supplied password	User authentication into the drive	SHA-384 Hash
KWK-BEV	256	Output of the PBKDF	Wrap/Unwrap the DEK	None
Unwrapped DEK	512	Direct output of the DRBG	User Data encryption/ decryption	None
Wrapped DEK	640	Initial device configuration	Persistent protection of the DEK	TOE NAND Memory

Creation, destruction and use of keys correspond to the following drive security state changes.

Activation: Upon provisioning to the security enabled state, the authorization factor is input to the PBKDF function along with a randomly generated 256 bit salt to produce the KWK (BEV). A salted hash of the authentication factor is also created. At that point the authorization factor is no longer needed and overwritten with zeros. The TOE's internal DRBG is invoked two additional times, obtaining the two 256 bit halves of the 512 bit DEK key (for use with the underlying XTS-AES-256). The KWK is then used to wrap the DEK and the KWK and DEK are no longer needed are overwritten by zeros. The new wrapped DEK, salted hash of the authorization factor, and other internal system configuration data is written into the newly allocated special NAND block. The drive is then reset which places it into the login state (awaiting user authentication).

Login: A salted hash of the authentication factor is created using the previously stored salt and compared against the previously stored value to confirm the correct authorization factor. If confirmed, the authorization factor and salt are input to the PBKDF function to produce the KWK. At that point the authorization factor is no longer needed and is overwritten with zeros. The salt is retained in SRAM in case of a change password command. The wrapped DEK is then unwrapped using the KWK and placed in the controller AES HW register. At this point the KWK is no longer needed and is overwritten by zeros. PCIe/NVMe drives also overwrite the DEK in SRAM while SATA models retain the DEK in SRAM due to internal memory management constraints.

Logout: The DEK in SRAM is overwritten by repeated value of 0x55 and the AES HW register is overwritten with zeros. All SRAM content including keys are destroyed when the device is powered off which corresponds to the only supported compliant power saving state (D3).

Change Password: When given a password change command, a new salted hash of the authorization factor is created. A new KWK is created via the PBKDF function and used to wrap the existing plaintext DEK. The salt used in the PBKDF is not erased or removed, and a new salt is not generated. User data is not physically erased however the TOE does revert to the Login state and all SRAM key information is overwritten with zeros. The previous user inaccessible NAND block is erased entirely by a block erase command, and a new block is allocated and the new keys, authorization factor hash, and TOE system information is saved. Because of this there is no read-verification of individual keys following the erase.

Zeroization: During a zeroization or a key destruction procedure, the TOE does not respond to any additional host commands. All keys, and key material, including the salt, are destroyed during this process by overwriting by zeros. After completion of the zeroize or key destruction action, all user data including volume & partition information is removed from the TOE using a NAND block erase, so the user is required to recreate the volumes and partitions on the drive before use. The TOE subsequently creates a new DEK for uninitialized/unprotected state operation.

The immediate destruction of valid key data is summarized in Table 9 below. Note memory locations may subsequently be overwritten by other values, however, these other values are not "destroying" the key and therefore are not included in FCS_CKM.4.

Table 9: Keychain Destruction

Security Function	State Before Command	State After command	Authorization factor (SRAM)	KWK-BEV (SRAM)	Unwrapped DEK (SRAM)	Unwrapped DEK (AES Engine HW Register)	Wrapped DEK (NAND)
Activate	Un-init	Login	Overwrite of 0x00	Overwrite of 0x00	Overwrite of 0x00	N/A	N/A
Login	Login	User	Overwrite of 0x00	Overwrite of 0x00	N/A	N/A	N/A
Logout	User	Login	N/A	N/A	Overwrite of 0x55	Overwrite of 0x00	N/A
Change PW	User	Login	Overwrite of 0x00	Overwrite of 0x00	Overwrite of 0x00	Overwrite of 0x00	Block Erase
Zeroize	User or Login	Un-init	N/A	N/A	Overwrite of 0x00	Overwrite of 0x00	Block Erase

FCS_CKM.1(b), FCS_CKM.1(c), FCS_CKM.4(a), FCS_CKM.4(b), FCS_CKM.4(d), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), FCS_CKM_EXT.6, FCS_KYC_EXT.1, FCS_KYC_EXT.2

7.1.3 Cryptographic Operations

The TOE itself implements and utilizes the following cryptographic operations to perform TSF:

Table 10: Cryptographic Operations

SFR	Algorithm	Description
FCS_COP.1(a)	ECDSA	Using a P-384 curve size and SHA-384 the TOE performs Digital Signature Verification to validate new firmware prior to installation.
FCS_COP.1(b)	SHA-256, SHA-384	The module implements SHA-256 with a block size of 512, and SHA-384 with a block size of 1024. The SHA-256 function is used in the Hash DRBG as well as used as the hashing function in the HMAC portion of the PBKDF. The SHA-384 implementation is used for firmware image integrity check, validation of the BEV, and ECDSA signature verification.
FCS_COP.1(c)	HMAC-SHA-256	Used in SP800-132 PBKDF: 256-bit key, SHA-256 hash, 512-bit block size, with an output length of 256-bits.
FCS_COP.1(d)	AES Key Wrap	AES Key Wrap operations use a 256-bit key and key wrap mode as specified in ISO/IEC 18033-3 NIST SP 800-38F. Using AES-KW, the

		TOE itself wraps and unwraps keys stored persistently in non-volatile memory.
FCS_COP.1(f)	XTS-AES-256	The user data on the disk is symmetrically encrypted using XTS-AES, as specified in IEEE 1619, with a pair of 256-bit keys.
FCS_KDF_EXT.1	PBKDF	SP800-132 PBKDF Function using HMAC-SHA-256. The PBKDF Function uses 1000 iterations to output a 256-bit key.
FCS_RBG_EXT.1	DRBG	SP 800-90A Hash DRBG using SHA-256.

The firmware update process utilizes ECDSA signature verification with a P-384 curve. The TOE includes a hard-coded hash value of the public key and is not subject to key destruction requirements. A full description of the process and use of these cryptographic algorithms is contained in 7.4.4 Trusted Update.

The TOE's internal NDRNG 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 a conditioning component utilizes the SHA-256 function in order to uniformly distribute entropy and provide a full 256-bit entropy output. This is used to seed (256-bit entropy input and 128 bit nonce) a SP800-90A Hash DRBG which is then used to generate the salt, keys, and all other key material used for the TSF. The NDRNG and DRBG also perform all SP800-90A/B health tests. The TOE does not require configuration of the RNGs.

The AES XTS tweak value is not randomly generated, but instead utilizes the Data Unit Sequence Number by combining NAND flash physical address information (bank * block * page) ranging between 0 and 4,294,967,296.

The TOE utilizes its internal DRBG to generate a 256-bit salt that is used as an input into the PBKDF2 function during user authentication. This salt, once generated, is stored in the TOE's internal NAND memory. During user authentication the salt is provided to the PBKDF2 function to generate the KWK or BEV.

FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(f), FCS_KDF_EXT.1, FCS_RBG_EXT.1, FCS_SNI_EXT.1, FPT_KYP_EXT.1

7.2 User Data Protection

7.2.1 Protection of Data on Disk

The TOE design is based on a single ASIC controller which includes a built-in hardware AES encryption engine and controls all memory components directly as shown in Figure 1 of section 1.4.1. There is no direct host user access to the SRAM. The TOE receives data from the host platform or computing system through the SATA/PCIe connections.

After Provisioning and subsequent Power-on, Logout, or Password Change command the TOE enters the Login state (initialized but unauthenticated) where it presents a blank read-only "shadow disk" from volatile DRAM memory. Once the user authenticates, the TOE enters the User state where all data reads and writes from the host are automatically passed through the AES encryption engine. Host written data is stored persistently in the TOEs NAND memory without any special user intervention. All host accessible areas are encrypted, including MBR and partition tables. 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.

During the write process, the TOE receives a write command from the host machine or microcontroller with a Logical Block Address and the plaintext data payload provided via the SATA/PCIe connection to the SSD Controller (gray square in Figure 1). The TOE parses the command inside the SSD Controller, this includes updating the address in the TOE's mapping table and placing the plaintext payload data in the TOE's volatile DRAM (orange square in Figure 1). The plaintext data then goes though the AES engine for encryption, followed by a write to the TOE's NAND flash memory (green square in Figure 1).

When data is being overwritten from the host to the same Logical Block Address, the TOE's SSD controller updates the mapping table and allocates a new Physical Block Address. The encrypted old data is then removed as background garbage collection is performed, either later or immediately via the host's TRIM command.

During a read process, the TOE receives a read command from the host machine or microcontroller with a Logical Block Address via the SATA/PCIe connection to the SSD Controller. The TOE parses the command inside the SSD Controller and searches for the matching Physical Block Address in the TOE's mapping table. The TOE retrieves the data from the NAND flash memory and decrypts the data through the AES engine. The plaintext decrypted data is then temporarily located in the TOE's volatile DRAM before being sent to the host/microcontroller via the SATA/PCIe interface.

FDP_DSK_EXT.1

7.3 Security Management

7.3.1 Specification of Management Functions

The TSF shall be capable of performing the following management functions:

a) Cryptographically erasing the DEK and changing to a new DEK is done by triggering the hardware Secure Erase signal, sending Zeroize command, or triggered by exceeding the maximum password attempts count.
b) 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.
c) Initiate Firmware update process is invoked by using a dedicated firmware update tool which is released via the vendor support site. The process is described in section 7.4.4.
d) Zeroize user data triggered by the hardware Secure Erase signal.

The TSF allows the User to cryptographically erase the DEK by issuing a zeroize command, triggering a hardware erase signal, or exceeding the maximum password attempts count. This in turn immediately cryptographically erases all user data.

The TOE supports a change password service only in an authorized state after passing user authentication. The user can change their password using either the recommended admin software tool or using the direct ATA/NVM commands as detailed in the TOE User Guidance documents.

The TSF allows the User to update to newer firmware via a firmware update tool. The TOE accepts a firmware update request only after verifying upper version numbering, public key, and firmware image integrity as described in Trusted Update.

The TSF does not require management or configuration of the TOE's RNGs.

The TOE supports two distinct user roles; the User, and the Crypto Officer. The Crypto Officer is responsible for configuring the TOE prior to field deployment and is required to ensure authenticity and integrity of the TOE. The TOE, once configured by the Crypto Officer, shall be provided to User, and the User shall follow rules set forth in the guidance document referenced in Section 9. The TOE does not mandate that these roles be separated to different operators, and in some instances, may require the User to also be the Crypto Officer.

FMT_SMF.1, FMT_SMR.1

7.4 Protection of the TSF

7.4.1 Protection of Key and Key Material

As described in 7.1.2, the Data Encryption Key is stored in a user inaccessible section of non-volatile NAND memory, where it is stored AES key-wrapped by the KWK or BEV. The SHA-384 hash of the authorization factor is also stored there. No other keys within the key chain are stored in non-volatile memory.

FPT_KYP_EXT.1

7.4.2 Power Saving States

In the Evaluated Configuration, the TOE supports only the following Compliant power saving state: D3 (off). The TOE enters this Compliant power saving state upon request from an authorized user and in the event of a system shutdown. The TOE does not allow administrators or users to manage or configure the Compliant power saving states supported by the TOE.

FPT_PWR_EXT.1, FPT_PWR_EXT.2, FMT_MOF.1

7.4.3 TSF Testing

The TOE performs several self-tests to ensure proper operation of the TSF. NDRNG and DRBG Self-Tests are performed at power-on as well as before the DRBG is invoked. All other self-tests are performed at power-on only and are listed in Table 11.

The Firmware Image Verification self-test is only to verify the integrity of the firmware image at TOE power-on. The firmware verification performed during trusted update, is considered verification of the new firmware being installed, rather than a self-test in the context of this document.

Table 11: Self-Tests

TOE Self-Test	Power-On	Before Using	Test Description
Firmware Image Verification	Yes	N/A	After loading the firmware image for the TOE’s own operation, The TOE verifies the integrity of the full firmware image code by comparing the SHA-384 hash digest output from currently running firmware with the stored hashed value which is stored separately in an index block during a firmware installation or an update procedure.
NDRNG Repetition Count Test	Yes	Yes	Get noise source 1024 times and check repetition counts per SP800-90B. (C=8)
NDRNG Adaptive Proportion Test	Yes	Yes	Get noise source 512 times and count identical pattern per SP800-90B. (C=135)
HASH DRBG KAT	Yes	Yes	Performs KAT by combining Instantiation, Reseed, and Generate.

HASH DRBG Health Tests	Yes	Yes	Performs KAT for each function of Instantiation, Reseed, and Generate per SP800-90A section 11.3.
HMAC SHA-256	Yes	N/A	Performs Known-Answer Test with 1024 bits input and 256 bits output and a 256-bit key size.
SHA-256 KAT	Yes	N/A	Performs Known-Answer Test with 392 bits input and 256 bits output.
SHA-384 KAT	Yes	N/A	Performs Known-Answer Test with 384 bits input and 384 bits output.
PBKDF2	Yes	N/A	Performs KAT with known 64 bytes password input and 256 bits MK output.
Key Wrap Key KAT	Yes	N/A	Wrap with 248b input and 312b output. Unwrap with 256b input and 192b output.
AES-XTS Encrypt KAT	Yes	N/A	Encryption KAT with 256 bits input and output with two 256 bits keys.
AES-XTS Decrypt KAT	Yes	N/A	Decryption KAT 256 bits input and output with two 256 bits keys.
ECDSA (P-384)	Yes	N/A	Perform signature verification and confirm the result by using 384 bits vector and keys.

FPT_TST_EXT.1

7.4.4 Trusted Update

The TOE supports updates to the device's internal firmware. The vendor releases a firmware update tool to each authorized user who authenticates their identity by logging into the vendor support site. The firmware update is signed and distributed by the manufacturer, Novachips. Based on NIST P-384 curve and domain parameter, the vendor generated an ECDSA private key and public key pair. While controlling confidentiality of the private key per internal regulation, the vendor prepares a firmware update tool package which consists of an installer, updated firmware image, digital signature, and public key. The process is diagrammed below.

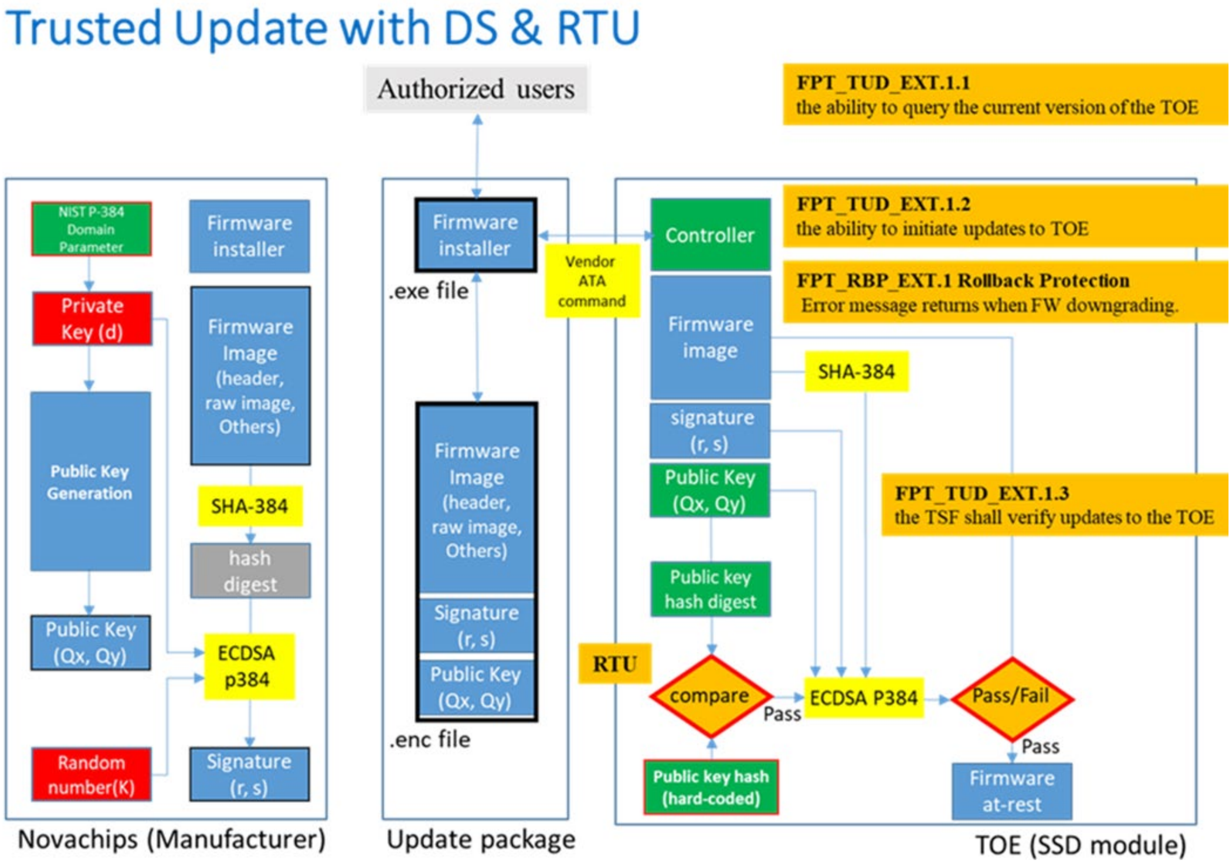

Figure 3: Trusted Update Flow

When the user initiates the firmware update process, the TOE firmware stored in write-protected storage verifies the below items before updating the firmware image:

- The authority of the user by allowing the firmware update process only in the User state after logging in with the correct password. A “Not user state” error message is displayed if the TOE is not in the user state.
- Firmware image signature public key by comparing its SHA-384 hash to the Root of Trust for Update (RTU) which is a reference hash of the public key stored in the ASIC controller one-time-programmable fuses during the production stage by the vendor. An “Invalid firmware image” error message is presented if the public key hash output is not identical with the hard-coded one.
- The integrity of the firmware image by performing ECDSA digital signature verification (decrypted signature matches calculated hash) using the verified public key. An “Invalid firmware image” error message is presented if the signature verification fails.
- The compliance of the firmware version control policy or rollback prevention by checking the version of the update image to the current version of the firmware. If the update image has a lower version the update tool shows the error message “Backward update error” and does not proceed with the update.

Upon completion of the firmware update, the previous firmware is erased, the TOE is required to be powered-off, and will utilize the new firmware version upon next power-on.

FPT_FAC_EXT.1, FPT_FUA_EXT.1, FPT_TUD_EXT.1, FPT_RBP_EXT.1

8. Terms and Definitions

Table 12: cPP Glossary

Term	Description
Authorization Factor	A value that a user knows, has, or is (e.g. password, token, etc) submitted to the TOE to establish that the user is in the community authorized to use the hard disk and that is used in the derivation or decryption of the BEV and eventual decryption of the DEK. Note that these values may or may not be used to establish the particular identity of the user.
Assurance	Grounds for confidence that a TOE meets the SFRs [CC1].
Border Encryption Value	A value passed from the AA to the EE intended to link the key chains of the two components.
Key Sanitization	A method of sanitizing encrypted data by securely overwriting the key that was encrypting the data.
Data Encryption Key (DEK)	A key used to encrypt data-at-rest.
Full Drive Encryption	Refers to partitions of logical blocks of user accessible data as managed by the host system that indexes and partitions and an operating system that maps authorization to read or write data to blocks in these partitions. For the sake of this Security Program Definition (SPD) and cPP, FDE performs encryption and authorization on one partition, so defined and supported by the OS and file system jointly, under consideration. FDE products encrypt all data (with certain exceptions) on the partition of the storage device and permits access to the data only after successful authorization to the FDE solution. The exceptions include the necessity to leave a portion of the storage device (the size may vary based on implementation) unencrypted for such things as the Master Boot Record (MBR) or other AA/EE pre-authentication software. These FDE cPPs interpret the term “full drive encryption” to allow FDE solutions to leave a portion of the storage device unencrypted so long as it contains no protected data.
Intermediate Key	A key used in a point between the initial user authorization and the DEK.
Host Platform	The local hardware and software the TOE is running on, this does not include any peripheral devices (e.g. USB devices) that may be connected to the local hardware and software.
Key Chaining	The method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key which encrypts the data; this method can have any number of layers.
Key Encryption Key (KEK)	A key used to encrypt other keys, such as DEKs or storage that contains keys.
Key Material	Key material is commonly known as critical security parameter (CSP) data, and also includes authorization data, nonces, and metadata.
Key Release Key (KRK)	A key used to release another key from storage, it is not used for the direct derivation or decryption of another key.
Operating System (OS)	Software which runs at the highest privilege level and can directly control hardware resources.
Non-Volatile Memory	A type of computer memory that will retain information without power.
Powered-Off State	The device has been shutdown.
Protected Data	This refers to all data on the storage device with the exception of a small portion required for the TOE to function correctly. It is all space on the disk a user could write data to and includes the operating system, applications, and user data. Protected data does not include the Master Boot Record or Pre-authentication area of the drive – areas of the drive that are necessarily unencrypted.
Submask	A submask is a bit string that can be generated and stored in a number of ways.
Target of Evaluation	A set of software, firmware and/or hardware possibly accompanied by guidance. [CC1]

Table 13: CC Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
AA	Authorization Acquisition
AES	Advanced Encryption Standard
BEV	Border Encryption Value
BIOS	Basic Input Output System
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with CBC-Message Authentication Code
CEM	Common Evaluation Methodology
CPP	Collaborative Protection Profile
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EE	Encryption Engine
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIPS	Federal Information Processing Standards
FDE	Full Drive Encryption
FFC	Finite Field Cryptography
GCM	Galois Counter Mode
GPIO	General Purpose Input/Output
HMAC	Keyed-Hash Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITSEF	IT Security Evaluation Facility
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IV	Initialization Vector
KEK	Key Encryption Key
KMD	Key Management Description
KRK	Key Release Key
MBR	Master Boot Record
MCU	Microcontroller Unit
NIST	National Institute of Standards and Technology
OS	Operating System
RBG	Random Bit Generator
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Algorithm
RTU	Root of Trust for Update
SAR	Security Assurance Requirement
SED	Self Encrypting Drive
SHA	Secure Hash Algorithm
SFR	Security Functional Requirement
SPD	Security Problem Definition
SPI	Serial Peripheral Interface
ST	Security Target
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Functionality
TSS	TOE Summary Specification

Table 13: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
USB	Universal Serial Bus
XOR	Exclusive or
XTS	XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing

9. References

Table 14: TOE Guidance Documentation

Reference	Description	Version	Date
[1]	Non-Proprietary Administrative Guidance	V1.1	January 20, 2023
[2]	ATA/NVM Command Guidance	V1.0	March 3rd, 2022

Table 15: Common Criteria v3.1 References

Reference	Description	Version	Date
[3]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model CCMB-2017-04-001	V3.1 R5	April 2017
[4]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components CCMB-2017-04-002	V3.1 R5	April 2017
[5]	Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components CCMB-2017-04-003	V3.1 R5	April 2017
[6]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2017-04-004	V3.1 R5	April 2017

Table 16: Supporting Documentation

Reference	Description	Version	Date
[7]	collaborative Protection Profile for Full Drive Encryption - Encryption Engine	2.0E	February 1, 2019
[8]	Supporting Document Mandatory Technical Document, Full Drive Encryption: Encryption Engine	2.0E	February 1, 2019
[9]	collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition	2.0E	February 1, 2019
[10]	Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition	2.0E	February 1, 2019
[11]	CC and CEM addenda for Exact Conformance, Selection-Based SFRs, Optional SFRs	0.5	May 2017