SSD Drives, Firmware Versions:
SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0

Security Target

Version 1.2

January 2025

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Description
1.0	15 Nov 2024	Released for check out
1.1	09 Jan 2025	Addressed ECR comments.
1.2	31 Jan 2025	Modified Non-TOE Hardware Components

Table of Contents

1 Introduction 4

1.1 Overview 4
1.2 Identification 4
1.3 Conformance Claims 4
1.4 Terminology 5

2 TOE Description 8

2.1 Type 8
2.2 Usage 8
2.3 Security Functions / Logical Scope 8
2.4 Physical Scope 9

3 Security Problem Definition 12

3.1 Threats 12
3.2 Assumptions 13
3.3 Organizational Security Policies 14

4 Security Objectives 15

5 Security Requirements 16

5.1 Conventions 16
5.2 Extended Components Definition 16
5.3 Functional Requirements 17
5.4 Assurance Requirements 25

6 TOE Summary Specification 26

6.1 Cryptographic Support (FCS) 26
6.2 User Data Protection (FDP) 29
6.3 Security Management (FMT) 30
6.4 Protection of the TSF (FPT) 30

7 Rationale 32

7.1 Conformance Claim Rationale 32
7.2 Security Objectives Rationale 32
7.3 Security Requirements Rationale 32

List of Tables

Table 1: Evaluation identifiers 4
Table 2: NIAP Technical Decisions 4
Table 3: Terminology 5
Table 4: CAVP Certificates 8
Table 5: TOE Hardware / Firmware 9
Table 6: Threats 12
Table 7: Assumptions 13
Table 8: Security Objectives for the Operational Environment 15
Table 9: Extended Components 16
Table 10: Summary of Security Function Requirements (SFRs) 17
Table 11: Assurance Requirements 25
Table 12: Cryptographic Key Usage, Storage, and Destruction 27
Table 13: TSF Self-Tests 31

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the UD Info SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 The TOE provide for encryption and decryption of user data stored on Self-Encrypting Drives (SEDs).

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	UD Info SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0
Security Target	UD Info SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0 Security Target, v1.2
Key Management Description	UD Info SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0 Key Management Description, v1.0 (15 November 2024) **Note.** This is a proprietary document and not publicly available referred to as (KMD) in this document.

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Full Drive Encryption – Encryption Engine, v2.0 + Errata 20190201 (referenced within as CPP_FDE_EE)

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Applicability Rationale
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Applicable.
TD0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Applicable.
TD0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Applicable.

TD #	Name	Applicability Rationale
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	Not Applicable, the TOE is not a NAS device.
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	Not Applicable, the TOE does not claim FCS_CKM.4(d)
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	Not Applicable, affected SFR selection not selected.

1.4 Terminology

Table 3: Terminology

Term	Definition
AA	Authorization Acquisition
AES	Advanced Encryption Standard
BEV	Border Encryption Value
BIOS	Basic Input Output System
CC	Common Criteria
CMOS	Complementary Metal-Oxide Semiconductor
CPP	Collaborative Protection Profile
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
EE	Encryption Engine
FIPS	Federal Information Processing Standards
FDE	Full Drive Encryption
GUI	Graphical User Interface
HMAC	Keyed-Hash Message Authentication Code
HW	Hardware

Term	Definition
IEEE	Institute of Electrical and Electronics Engineers
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
IV	Initialization Vector
KEK	Key Encryption Key
KMD	Key Management Description
KW	Key Wrap
MBR	Master Boot Record
NIST	National Institute of Standards and Technology
NVMe PCIe	Non-Volatile Memory Express Peripheral Component Interconnect Express
OS	Operating System
OTP	One-Time Programmable
PBKDF	Password-Based Key Derivation Function
RBG	Random Bit Generator
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Algorithm
RTU	Root of Trust
SATA	Serial Advanced Technology Attachment
SED	Self-Encrypting Drive
SHA	Secure Hash Algorithm
SFR	Security Functional Requirements
SSD	Solid-State Drive
ST	Security Target
SPD	Security Problem Definition
TCG Opal	Trusted Computing Group Opal
TOE	Target of Evaluation

Term	Definition
TSF	TOE Security Functionality
TSS	TOE Summary Specification
XOR	Exclusive or
XTS	XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing

2 TOE Description

2.1 Type

4 The TOE is a self-encrypting drive that provides encryption and decryption of stored user data.

2.2 Usage

5 The TOE provides full drive encryption to safeguard data on lost or stolen devices. The Encryption Engine (EE) ensures that the drive data is encrypted using National Institute of Standards and Technology (NIST) approved algorithms. The operation of the TOE is transparent to users, who interact with the Authorization Acquisition (AA) component. The evaluation is limited to the Encryption Engine (EE) component only and Authorization Acquisition (AA) component is not part of this evaluation.

2.3 Security Functions / Logical Scope

6 The TOE provides the following security functions:

a) **Data Protection.** The TOE enables encryption and decryption of user data on a SED (Self-Encrypting Drive) to protect it from unauthorized disclosure.

b) **Secure Key Material.** The TOE ensures key material used for storage encryption is properly generated and protected from disclosure. It also implements cryptographic key and key material destruction during transitioning to a Compliant power saving state, or when all keys and key material are no longer needed.

c) **Secure Management.** The TOE enables management of its security functions, including:

  i) Changing and erasing the Data Encryption Key (DEK)

  ii) Updating the TOE firmware

d) **Trusted Update.** The TOE ensures the authenticity and integrity of firmware updates through digital signatures using Rivest Shamir Adleman Algorithm (RSA) 2048 with Secure Hash Algorithm (SHA)-256.

e) **Self-Testing.** The TOE ensures its integrity and operation by performing self-tests.

f) **Cryptographic Operations.** The TOE performs cryptographic operations as shown in Table 4, which includes relevant Cryptographic Algorithm Validation Program (CAVP) certificates.

Table 4: CAVP Certificates

SFR / Requirement	Capability	Certificate
FCS_COP.1(a) (Signature Verification)	RSA SigVer (FIPS 186-4)	C1356, C1358
FCS_COP.1(b) (Hash Algorithm)	SHA2-256	C1356, C1358

SFR / Requirement	Capability	Certificate
FCS_COP.1(c) (Message Authentication)	HMAC-SHA2-256	C1356, C1358
FCS_COP.1(f) (AES Data Encryption/Decryption)	AES-XTS-256	C1356, C1358
FCS_RBG_EXT.1 (Random Bit Generation)	HMAC-DRBG	C1356, C1358

2.4 Physical Scope

7 The physical boundary of the TOE encompasses the Solid-State Drive SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0 firmware running on the SEDs identified in Table 5. The TOE hardware is delivered to customers via trusted courier with the firmware preinstalled.

8 The TOE models support either Non-Volatile Memory Express Peripheral Component Interconnect Express (NVMe PCIe) or Serial Advanced Technology Attachment (SATA) III interfaces. All TOE models incorporate an ARM Cortex-R5 processor (ARMv7-R microarchitecture).

Table 5: TOE Hardware / Firmware

Drive	Hardware (HW) P/N	Controller	FW Version
2.5-inch SATA NAND Flash SSD	HF3-25DA128GB-A8P	PS3112-S12	SCPU13.0
	HF3-25DA256GB-A8P		
	HF3-25DA512GB-A8P		
	HF3-25DA001TB-A8P		
	HF3-25DA002TB-A8P		
	HF3-25DA256GB-B8P	PS3112-S12	SCQU15.0
	HF3-25DA512GB-B8P		
	HF3-25DA001TB-B8P		
	HF3-25DA002TB-B8P		
	HF3-25DA004TB-B8P		
M.2 2280 SATA NAND Flash SSD	M2S-80DA128GB-A8P	PS3112-S12	SCPU13.0
	M2S-80DA256GB-A8P		

Drive	Hardware (HW) P/N	Controller	FW Version
	M2S-80DA512GB-A8P		
	M2S-80DA001TB-A8P		
	M2S-80DA002TB-A8P		
	M2S-80DA256GB-B8P	PS3112-S12	SCQU15.0
	M2S-80DA512GB-B8P		
	M2S-80DA001TB-B8P		
	M2S-80DA002TB-B8P		
M.2 2280 NVMe NAND Flash SSD	M2P-80DA256GB-A8P	PS5012-E12	ECPU13.0
	M2P-80DA512GB-A8P		
	M2P-80DA001TB-A8P		
	M2P-80DA002TB-A8P		
	M2P-80DA256GB-BEP	PS5012-E12	ECQU15.0
	M2P-80DA512GB-BEP		
	M2P-80DA001TB-BEP		
	M2P-80DA002TB-BEP		

2.4.1 Guidance Documents

9 The TOE includes the following guidance documents:

a) UD Info SSD Drives, Firmware Versions: SCPU13.0/ECPU13.0/SCQU15.0/ECQU15.0 Common Criteria Guide, v1.1 (PDF).

2.4.2 Non-TOE Components

10 The TOE operates with the following components in the environment:

a) **Authorization Acquisition.** Trusted Computing Group Opal (TCG OPAL) v2.0 compliant PBA software installed on a 128 MB read-only Shadow Master Boot Record (MBR) partition on the SED. This is the AA component that supplies the Border Encryption Value (BEV) for locking and unlocking the drives. The AA software provides the Graphical User Interface (GUI) used for performing the security management functions described within this ST.

   i) Testing performed using KLC CipherDriveOne v2.0.1

b) **Protected OS.** The TOE supports protection of commonly used operating systems, such as Linux Operating Systems/Linux based Hypervisors and Windows Operating Systems.

c) **Computer Hardware.** Intel based UEFI booted systems that supports Intel Secure Key Technology. CC Testing performed using CPUs:

   i) Intel Core i5-13500 (Raptor Lake)

   ii) Intel Core i5-8400 (Coffee Lake)

2.4.3 Not included in the TOE Evaluation

11 The evaluation is limited to those security functions identified in section 2.3.

12 The following configuration has not been evaluated:

a) Use of multiple drives.

3 Security Problem Definition

13 The Security Problem Definition is reproduced from the CPP_FDE_EE.

3.1 Threats

Table 6: Threats

Identifier	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_MATERIA L_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.
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 DEKs 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 exhaust the key space through brute force 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.

Identifier	Description
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 that bypasses the intended security features and provides them unauthorized 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 Assumptions

Table 7: Assumptions

Identifier	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 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 unpartitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors, un-partitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE pre-authentication software) contain no protected data.

Identifier	Description
A.TRAINED_USER	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	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.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.

3.3 Organizational Security Policies

14 None defined.

4 Security Objectives

15 The security objectives are reproduced from the CPP_FDE_EE.

Table 8: Security Objectives for the Operational Environment

Identifier	Description
OE.TRUSTED_ CHANNEL	Communication among and between product components (i.e., 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	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.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.TRAINED_USERS	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.
OE.PHYSICAL	The Operational Environment will provide a secure physical computing space such than an adversary is not able to make modifications to the environment or to the TOE itself.

5 Security Requirements

5.1 Conventions

16 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with *italicized* text.

b) **Refinement.** Indicated with **bold** text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with *italicized and underlined* text.

e) **Iteration.** Indicated by appending parentheses that contain a letter that is unique for each iteration, e.g. (a), (b), (c) and/or with a slash (/) followed by a descriptive string for the SFR's purpose, e.g. /Server.

17 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the PP.

5.2 Extended Components Definition

18 The following Extended Components are defined in Appendix C.2 of the CPP_FDE_EE:

Table 9: Extended Components

Requirement	Title
FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
FCS_CKM_EXT.6	Cryptographic Key Destruction Types
FCS_KYC_EXT.2	Key Chaining (Recipient)
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FCS_VAL_EXT.1	Validation
FDP_DSK_EXT.1	Protection of Data on Disk
FPT_KYP_EXT.1	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update

Requirement	Title
Selection based	
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_RBG_EXT.1	Random Bit Generation
FPT_FUA_EXT.1	Firmware Update Authentication

5.3 Functional Requirements

Table 10: Summary of Security Function Requirements (SFRs)

Requirement	Title
FCS_CKM.1(c)	Cryptographic Key Generation (Data Encryption Key)
FCS_CKM.4(a)	Cryptographic Key Destruction (Power Management)
FCS_CKM_EXT.4(a)	Cryptographic Key and Key Material Destruction (Destruction Timing)
FCS_CKM_EXT.4(b)	Cryptographic Key and Key Material Destruction (Power Management)
FCS_CKM_EXT.6	Cryptographic Key Destruction Types
FCS_KYC_EXT.2	Key Chaining (Recipient)
FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
FCS_VAL_EXT.1	Validation
FDP_DSK_EXT.1	Protection of Data on Disk
FMT_SMF.1	Specification of Management Functions
FPT_KYP_EXT.1	Protection of Key and Key Material
FPT_PWR_EXT.1	Power Saving States
FPT_PWR_EXT.2	Timing of Power Saving States
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
Selection based	
FCS_CKM.1(b)	Cryptographic Key Generation (Symmetric Keys)
FCS_CKM.4(b)	Cryptographic Key Destruction (TOE-Controlled Hardware)

Requirement	Title
FCS_COP.1(a)	Cryptographic Operation (Signature Verification)
FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
FCS_COP.1(c)	Cryptographic Operation (Message Authentication)
FCS_COP.1(d)	Cryptographic Operation (Key Wrapping)
FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_KDF_EXT.1	Cryptographic Key Derivation
FCS_RBG_EXT.1	Random Bit Generation
FPT_FUA_EXT.1	Firmware Update Authentication

5.3.1 Cryptographic Support (FCS)

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

FCS_CKM.1.1(b) **Refinement:** The TSF shall generate **symmetric** cryptographic keys using **a Random Bit Generator as specified in FCS_RBG_EXT.1** and specified cryptographic key sizes [256 bit] that meet the following: [*no standard*].

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

FCS_CKM.1.1(c) **Refinement**: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation ~~algorithm~~ **method** [

- generate a DEK using the RBG as specified in FCS_RBG_EXT.1,]

and specified cryptographic key sizes [256 bits] ~~that meet the following: [*assignment: list of standards*]~~.

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

FCS_CKM.4.1(a) **Refinement:** The TSF shall **[erase] 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*].

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

FCS_CKM.4.1(b) **Refinement**: 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],**
- **removal of power to the memory];**

- **For non-volatile memory [**
  - **that employs a wear-leveling algorithm, the destruction shall be executed by a [**
    - **single overwrite consisting of zeroes,**
    - **overwrite with a new value of a key of the same size,**
    - **block erase];**

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

] that meets the following: [*no standard*].

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

FCS_CKM_EXT.4.1(a) The TSF shall destroy all keys and key material when no longer needed.

FCS_CKM_EXT.4(b) 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.

FCS_CKM_EXT.6 Cryptographic Key Destruction Types

FCS_CKM_EXT.6.1 The TSF shall use [FCS_CKM.4(b)] key destruction methods.

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

FCS_COP.1.1(a) **Refinement:** The TSF shall perform [*cryptographic signature services (verification)*] in accordance with a [

- **RSA Digital Signature Algorithm with a key size (modulus) of [2048-bit];**

]

that meet the following: [

- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 29 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA schemes]**

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

FCS_COP.1.1(b) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [**SHA-256**] ~~and cryptographic key sizes [*assignment: cryptographic key sizes*]~~ that meet the following [ISO/IEC 10118-3:2004].

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

FCS_COP.1.1(c) **Refinement:** The TSF shall perform cryptographic [*message authentication*] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256**] and cryptographic key sizes [***256 bits [HMAC]***] that meet the following: [ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"].

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

FCS_COP.1.1(d) **Refinement:** 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]**].

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

FCS_COP.1.1(f) **Refinement**: 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/IEC 18033-3,* ***[XTS as specified in IEEE 1619]***].

FCS_KDF_EXT.1 Cryptographic Key Derivation

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.

FCS_KYC_EXT.2 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): [

- symmetric key generation as specified in FCS_CKM.1(b),
- key derivation as specified in FCS_KDF_EXT.1,
- key wrapping as specified in FCS_COP.1(d)]

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

FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with [[*NIST SP 800-90A*]] using [HMAC_DRBG (any)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [

- [*1*] 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.

FCS_SNI_EXT.1 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].

FCS_VAL_EXT.1 Validation

FCS_VAL_EXT.1.1 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 The TSF shall require 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 [

- require power cycle/reset the TOE after [*an administrator configurable number between 1 and 10*] of consecutive failed validation attempts].

5.3.2 User Data Protection (FDP)

FDP_DSK_EXT.1 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.

5.3.3 Security Management (FMT)

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 **Refinement:** 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) **[*no other functions*]**].

5.3.4 Protection of the TSF (FPT)

FPT_FUA_EXT.1 Firmware Update Authentication

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.

FPT_KYP_EXT.1 Protection of Key and Key Material

FPT_KYP_EXT.1.1 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)].

FPT_PWR_EXT.1 Power Saving States

FPT_PWR_EXT.1.1 The TSF shall define the following Compliant power saving states: [D3].

FPT_PWR_EXT.2 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, [no other conditions].

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 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 integrity*
- *DRBG health*
- *Known Answer Tests (KATs):*
  - *AES XTS encrypt/decrypt*
  - *AES key wrap/unwrap*
  - *DRBG*
  - *SHA-2*
  - *HMAC*
  - *PBKDF2*

]

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 **Refinement:** The TSF shall provide [*authorized users*] the ability to query the current version of the TOE **[firmware]** ~~**software/firmware**~~**.**

FPT_TUD_EXT.1.2 **Refinement:** The TSF shall provide [*authorized users*] the ability to initiate updates to TOE **[firmware]** ~~**software/firmware**~~.

FPT_TUD_EXT.1.3 **Refinement:** The TSF shall verify updates to the TOE [**firmware**] using a [**authenticated firmware update mechanism as described in FPT_FUA_EXT.1**] by the manufacturer prior to installing those updates.

5.4 Assurance Requirements

19 The TOE security assurance requirements are summarized in Table 11.

Table 11: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - sample
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

20 In accordance with section 6.1 of the CPP_FDE_EE, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR, **including a proprietary Key Management Description (Appendix E), and [Entropy Essay].**

6 TOE Summary Specification

6.1 Cryptographic Support (FCS)

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

21 The TOE generates the Data Encryption Key (DEK) using the *Change DEK* option in the GUI. The process invokes the internal HMAC_DRBG when generating the DEK.

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

22 The TOE generates a 256-bit AES DEK which is protected by the Key Encryption Key (KEK) using the key wrap function.

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

23 The TOE erases cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state. All keys in the chain (DEK, KEK, and BEV) are erased from volatile memory by performing a single overwrite consisting of zeroes.

24 For non-volatile memory, the DEK is erased in two stages. First, the old key is overwritten with the new key value and then stored in a new location in memory. The old block location (where the original key was stored) is erased using a wear-leveling program. User KEKs are erased from non-volatile memory by performing a single overwrite consisting of zeroes. The BEV is not stored in non-volatile memory.

25 Additional information on key usage, storage, and destruction can be found in Table 12 below.

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

26 The following table describes the how cryptographic keys are used, stored and destroyed.

Table 12: Cryptographic Key Usage, Storage, and Destruction

Key	Key Type/Length	Initialization	Usage	Storage	Destruction	Destruction Timing
DEK	XTS-AES-256	TOE configuration	Data encryption/ decryption	Encrypted by KEK and stored in NAND. Plaintext keys stored in DRAM and registers.	Replaced by new key followed by a Block erase. Zeroization	All keys are destroyed when the following occurs: • When a user password change occurs • After the user session ends • After a power off • When the TOE is uninstalled • When the Change DEK option is executed via the GUI
KEK	AES Key Wrap 256	TOE configuration	Protected, wrapped DEK	Encrypted by user password-based key with AES key wrap and stored in NAND. Plaintext keys stored in DRAM and registers.	Zeroization	
BEV	PBKDF	Output of PBKDF	Unwrap of KEK	Plaintext keys stored in DRAM and registers.	Zeroization	

27 **Note**: The TOE includes both volatile memory (DRAM) and non-volatile memory (NAND). In both cases, the memory is accessed using standard microcontroller memory interface controllers and addressing schemes. The DRAM is bit-level addressable, and NAND flash is block-level readable and writable. The TOE does not persistently store plaintext keys. Only protected keys and copies are persistently stored in NAND with parity bits. All protected keys used for the microcontroller are stored in a single block of NAND that is inaccessible to the host.

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

28 Details regarding the timing of key and key material destruction can be found in Table 12.

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

29 Details regarding key destruction when entering a Compliant power saving state are provided in sections 6.1.3 and 6.1.4 above.

6.1.7 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

30 All keys are destroyed as per the methods described in FCS_CKM.4(b). The TOE's key chain is described in the Key Management Description (KMD).

6.1.8 FCS_COP.1(a) and FCS_COP.1(b) Cryptographic Operation (Signature Verification and Hash Algorithm)

31 All firmware binaries are signed by Phison. Phison is the primary developer of the TOE firmware and is the only authorized source for code signing.

32 The TOE performs signature verification using RSA 2048 with SHA-256 for trusted updates as follows:

a) TOE updates are signed with the code signing private key.

b) The obfuscated public key is embedded in the TOE binary.

c) When the user triggers the TOE update, the TOE compares a hash of the public key with the stored hash of the public key, and then verifies the digital signature.

d) If the digital signature verification succeeds, the upgrade process is carried out.

e) If the digital signature verification fails, the upgrade process is aborted, and an error is displayed to the user.

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

33 The TOE implements HMAC-SHA-256 with the following characteristics:

a) **Key length.** 256 bits.

b) **Block size.** 512 bits.

c) **MAC length.** 256 bits.

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

34 The TOE key wrap function is used to protect the DEK using AES-256.

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

35 The TOE performs data encryption/decryption using AES-XTS with 256-bit keys.

6.1.12 FCS_KDF_EXT.1 Cryptographic Key Derivation

36 Passwords are conditioned via PBKDF2 using HMAC-SHA-256 with 1,000 iterations, resulting in a 256-bit key in accordance with NIST SP 800-132.

6.1.13 FCS_KYC_EXT.2 Key Chaining (Recipient)

37 The TOE key chain is described in the KMD.

6.1.14 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

38 The TOE uses a hardware-based deterministic random bit generator (DRBG) that complies with NIST SP 800-90A for all cryptographic operations. The DRBG is seeded with at least 256-bits of entropy from thermal noise generated by the Complementary Metal-Oxide Semiconductor (CMOS).

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

39 The TOE generates 32 byte salts using RAND_bytes at the time of encryption which are then stored in a database for use during decryption. Salts are generated using the DRBG as described in FCS_RBG_EXT.1.

40 Unique 16 byte nonces are generated using the hardware Random Number Generator (RNG) and are appended to the encrypted data.

41 The Logical Block Address (LBA) of the SSD is used as the tweak value. Tweak values are non-negative integers, assigned consecutively, and start at an arbitrary non-negative integer. The tweak value is converted to a little-endian byte array, where encryption of the tweak is done using AES-XTS.

6.1.16 FCS_VAL_EXT.1 Validation

42 The TOE will validate a BEV using key wrap as specified in FCS_COP.1(d). As per the application note in the CPP_FDE_PP, when the key wrap in FCS_COP.1(d) is used, the validation is performed inherently.

43 Successful validation of the BEV per above is required prior to allowing decryption of the drive and granting access to any TSF data after exiting a compliant power saving state.

44 After a configurable number of failed authentication attempts is reached, the system will lockout the user account and stop responding until it is rebooted at which point the lockout counter is reset. An administrator can set this threshold to a value between 1 and 10 failed attempts.

6.2 User Data Protection (FDP)

6.2.1 FDP_DSK_EXT.1 Protection of Data on Disk

45 The first 128MB of media data on the drive (Shadow MBR data) and the disk partition tables are read only and not encrypted. Once provisioned, all other data written to disk is encrypted without user intervention using AES-XTS. Data being written to disk is encrypted before being programmed to NAND storage.

46 The following initialization activities ensure the encryption function works when first provisioning the drive:

a) Examine the tamper evidence and check the module has not been tampered.

b) StartSession SID of AdminSP with MSID password, and then set new password for SID password. The new password shall be at least 20 bytes.

c) Disable AdminSP “Makers” Authority.

d) Execute TCG activate command to have the module enter TCG active mode.

e) StartSession Admin1 of LockingSP with new password of SID in Step2, and then set new password for Admin1-4 passwords and User1-9 passwords of LockingSP. The new passwords shall be at least 20 bytes.

f) Configure all LockingRanges of LockinSP by setting ReadLockEnabled and WriteLockEnabled columns to TRUE.

g) Power cycle the module.

h) Check if the module is in the FIPS approved mode by using the Identify command response data byte 506 bit1 (SATA) or the Identify controller command response data byte 4093 bit1 (NVMe). The bit1 shall be set to 1.

i) Check the module's firmware version using the Identify command response data dword 23-26 (SATA) or the Identify controller command response data byte 64-71 (NVME).

47 Once provisioned, the following boot initialization process is performed each time the TOE transitions from a power saving state:

a) CTL ROM code conducts KAT of SHA-256bit/RSA 2048 bit as listed in FPT_TST_EXT.1, Table 13.

b) FW code is loaded from NAND.

c) CTL ROM code conducts firmware integrity check of the FW binary via RSA 2048 SHA256 PSS Signature Verification.

d) FW code is executed (only if integrity check is successful).

e) FW code conducts all firmware power-on self-test as listed in FPT_TST_EXT.1, Table 13.

f) When all self-tests have passed, the module enters a ready state awaiting host/use commands.

6.3 Security Management (FMT)

6.3.1 FMT_SMF.1 Specification of Management Functions

48 The DEK can only be changed by generating a new one. DEKs are generated by using the *Change DEK* option via the GUI. DEKs are erased as per FCS_CKM.4(a) described in section 6.1.3 above.

49 Users of the TOE must contact the vendor to obtain firmware updates. Firmware updates are manually installed by authorized administrators.

6.4 Protection of the TSF (FPT)

6.4.1 FPT_FUA_EXT.1 Firmware Update Authentication

50 Firmware running on the TOE exists in ROM. The Root of Trust (RTU) uses a SHA-256 hash of the public key, as specified in FCS_COP.1(b), to authenticate the source of firmware updates. The public key hash is stored in One-Time Programmable (OTP) memory. The ROM code loads the firmware update and checks the hash of the public key embedded in the firmware binary. ROM code is hardcoded in the controller hardware and cannot be modified post-production.

6.4.2 FPT_KYP_EXT.1 Protection of Key and Key Material

51 Keys stored in non-volatile memory are wrapped, as specified in FCS_COP.1(d).

6.4.3 FPT_PWR_EXT.1 Power Saving States

52 The TOE supports the following Compliant power saving states:

a) **D3**. Powered Off – user initiated.

6.4.4 FPT_PWR_EXT.2 Timing of Power Saving States

53 The TOE enters a Compliant power saving state as prompted by the protected Operating System (OS) when user-initiated requests as described in Section 6.4.3 above.

6.4.5 FPT_TST_EXT.1 TSF Testing

54 Table 13 below defines the self-tests performed by the TOE during initial start-up (power-on).

Table 13: TSF Self-Tests

Self Test	Description
Rom Code SHA 256 bit	KAT
Rom Code RSA 2048 bit	KAT
Boot Loader Integrity	Firmware integrity test
Firmware AES XTS 256 bit Encrypt	KAT
Firmware AES XTS 256 bit Decrypt	KAT
Firmware SHA 256 bit	KAT
Firmware HMAC SHA 256	KAT
Firmware AES Key Wrap	KAT
Firmware AES Key Unwrap	KAT
Firmware DRBG	KAT
Firmware DRBG Health Tests	SP 800-90A Section 11.3 Health Tests
Firmware SP 800-132 PBKDF2	KAT
DRBG	Continuous RNG test for DRBG
NDRNG	Continuous RNG test for NDRNG

6.4.6 FPT_TUD_EXT.1 Trusted Update

55 Update files are digitally signed by Phison and verified prior to installation using the authenticated firmware update mechanism described in FPT_FUA_EXT.1. Update files contain a digital signature that is embedded within the binary. Additional process details are described in Section 6.4.1 above.

7 Rationale

7.1 Conformance Claim Rationale

56 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is consistent with the CPP_FDE_EE.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the CPP_FDE_EE.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the CPP_FDE_EE.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the CPP_FDE_EE. No additional requirements have been specified.

7.2 Security Objectives Rationale

57 All security objectives are drawn directly from the CPP_FDE_EE.

7.3 Security Requirements Rationale

58 All security requirements are drawn directly from the CPP_FDE_EE. No optional SFRs are included in the ST. The following selection based SFRs have been included:

a) FCS_CKM.1(b)

b) FCS_CKM.4(b)

c) FCS_COP.1(a)

d) FCS_COP.1(b)

e) FCS_COP.1(c)

f) FCS_COP.1(d)

g) FCS_COP.1(f)

h) FCS_KDF_EXT.1

i) FCS_RBG_EXT.1

j) FPT_FUA_EXT.1