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

Version: 1.1

2020-2-06

Prepared For:

Mercury Systems, Inc.

3601 E University Dr

Phoenix, AZ 85034

Prepared By:

Devin Becker

UL Verification Services Inc.

Document Change Log

Version	Date	Author	Changes
1.0	1/8/2020	Devin Becker	Original Document
1.1	2/06/2020	Devin Becker	Updated to address ECR comments

Table of Contents

1. Security Target (ST) Introduction

- The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
- The ST reference shall uniquely identify the ST.
- The TOE reference shall identify the 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, Target of Evaluation (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:	Security Target for Mercury Systems ASURRE-Stor™ Solid State Self-Encrypting Drive
ST Version Number:	Version 1.1
ST Author(s):	Devin Becker
ST Publication Date:	2-06-2020
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	Mercury Systems, Inc. 3601 E University Dr Phoenix, AZ 85034
TOE Name:	ASURRE-Stor™ Solid State Self-Encrypting Drive
TOE Version	Hardware revision 3.0, Firmware revision 1.5.1

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is the Mercury Systems ASURRE-Stor™ Solid State Self-Encrypting Drive.

1.3.2 TOE Usage

The TOE functions as a standard 2.5" SATA self-encrypting solid state hard drive. The TOE is a solid state device that stores all user data in encrypted form. This provides highly secure storage of data and facilitates rapid cryptographic erasure via sanitization of the encryption key.

1.3.3 TOE Major Security Features Summary

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

1.3.4 TOE IT environment hardware/software/firmware requirements

The physical embodiment conforms to the EIA SFF-8201 specification. The electrical and software interface is the Serial ATA revision 2.6 specification. As such it can interface to any environment that is compatible with standard 2.5" SATA hard drives. The TOE also uses two of the SATA power interface lines as a serial interface that can serve as an optional method of entering the Key Chain parameters when in KEK with Black Key mode. The TOE also has optional status LEDs and a Write Protect Port. The TOE can utilize the industry standard ATA security functions to authenticate users and can load or generate its own encryption keys and as such is not dependent on TCG based hardware or a TPM module. External software and hardware capable of sending and receiving ATA commands is required for operation. Optional external software and hardware can be used to load keys via the serial interface. The TOE developer provides an optional PC software utility with a user friendly graphical user interface for configuration called the MDU[1].

1.4 Target of Evaluation Description

1.4.1 Target of Evaluation Physical Boundaries

The TOE consists of firmware revision 1.5.1 and hardware revision 3 of the following models:

- ASD256AM2R
- ASD512AM2R

  These two models are identical except for the amount of NAND memory onboard.

- ADR256AM2R
- ADR512AM2R

  These two models are identical except for the amount of NAND memory onboard.

[1] The use of the MDU is optional, but was not evaluated as part of the Common Criteria certification process.

> The ADR and ASD models differ only in how much NAND memory is reserved for use as bad cell replacement memory. It is not accessible to the user. For example, the ADR256 has 240 GB of user-addressable space, while the ASD256 has 200 GB of user-addressable space. The difference is held in reserve by the TOE firmware for use when NAND cells are determined to be worn out and must be replaced.

The physical boundary of the TOE is the drive enclosure. The main processor is an Altera NIOS II, which is a CPU and FPGA. The programmed FPGA is referred to as the Armor Processor.

The use of the MDU is optional, but was not evaluated as part of the Common Criteria certification process.

The guidance documentation that is part of the TOE is listed in Section 9, "References," within Table 15: TOE Guidance Documentation.

The required documents are provided to the consumer in .pdf format and are available using Mercury's SSH File Transfer Protocol (SFTP) server after an account has been created.

1.4.2 Target of Evaluation Logical Boundaries

The TOE offers the following logical security features:

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

As specified in FMT_SMF.1.1 below, the TOE can be configured to operate in one of several modes of operation. The modes of operation differ in how the TOE DEK is established and stored. The following modes are covered by this evaluation:

- KEK with Black Key (manual encrypted DEK entry) with ATA password.
- Permanent Key (self-generated key, AES-wrapped with a user-supplied password)

The TOE can be configured for the following modes but they are NOT covered by this evaluation:

- Session key (manual plain-text key entry) with and without ATA password.
- Permanent key (manual plain-text key entry) with and without ATA password.
- Permanent Key (self-generated) without ATA password.
- KEK with Black Key (manual encrypted DEK entry) without ATA password.

1.4.3 TOE Description

The logical boundary of the TOE include 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 these security functions are provided in Section 7, "TOE Summary Specification."

1.4.3.1 Cryptographic Support

The drive utilizes the following cryptographic algorithms that are approved for use by FIPS 140-2 Annexes A, C, and D. The TOE uses a FIPS 140-2 validated module (certificate #2884).

Table 1: FIPS Approved Cryptographic Algorithms			
CAVP Cert	Algorithm	Standard	Use
2802, 3987	XTS-AES-256	FIPS 197, SP 800-38E	Used for primary data storage
1179	DRBG	SP 800-90A	Used in key generation
3986	AES Key Wrap	SP 800-38F	Used for all key storage
3291	SHA-512	FIPS 180-4	Used for hashing function in HMAC and DRBG
2602	HMAC	FIPS 198-1	Message authentication for passwords
V.A.	PBKDF	SP 800-132	Option 2a for protecting data encryption key
883	ECDSA	FIPS 186-4	Used for Firmware upgrade

1.4.3.2 User Data Protection

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

1.4.3.3 Security Management

The TOE allows authorized users to change the data encryption key (DEK), cryptographically erase the DEK, initiate firmware updates, import wrapped DEK, change passwords, and configure cryptographic functionality.

1.4.3.4 Protection of the TSF

The TOE protects itself by running a suite of self-tests at power-up, authenticating firmware and by not providing any mechanism to export any key values. The customer is encouraged to externally fill keys so that an unpowered module contains no CSP information that would lead to compromise of the encrypted data at rest. Beyond self-tests and crypto KATs, the module has numerous continuously running checks built into the C code and the VHDL code. Whenever an error is detected, (corruption, impossible states, out of range values, extra bytes in queues, etc.) that might compromise the security of the module, the module sets a flag and resets. This eliminates any CSP values in FPGA RAM and renews/reloads logic in the FPGA.

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.

Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification. Those notes specific to the TOE are marked “TOE Application Note;” those taken from the FDE Protection Profile are marked “PP Application Note.”

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 notation used in those PP 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. Additionally, obvious errors in the cPP are corrected and noted as such.

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.

Iterations performed by the ST Author are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1).

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 <u>underlined text</u>**.**

Refinements that add text use ***bold and italicized text*** to identified the added text***.*** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

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 [4][3], and CC Part 3 extended [5].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0 + Errata 20190201, February 1, 2019 and the collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0 + Errata 20190201, February 1, 2019. These Protection Profiles will be referred to individually or collectively as FDE or cPP for convenience throughout this Security Target.

Table 2: Applied Technical Decisions

TD	TD Title
cPP_FDE_AA_V2.0E Technical Decisions	
0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities
cPP_FDE_EE_V2.0E Technical Decisions	
0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities
0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states
0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states

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

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
    - All threats defined in the cPP are carried forward to this ST;
    - No additional threats have been defined in this ST.
- Organizational Security Policies
    - All OSP defined in the cPP are carried forward to this ST;

o No additional OSPs have been defined in this ST.

- Assumptions
    - All assumptions defined in the cPP are carried forward to this ST;

    - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
    - All objectives defined in the cPP are carried forward to this ST.
- All SFRs and SARs defined in the cPP are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the cPP have been properly instantiated in this Security Target; therefore, this ST shows exact compliance 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[2]	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 key material of equal importance to the data itself. Threat agents may look for key material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash.
T.KEYING_MATERIAL_ COMPROMISE[3]	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[4]	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 otherwise put it in a state in which it discloses protected data to unauthorized users.
T.AUTHORIZATION_GUESSING[5]	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

[2] T.KEYING_MATERIAL_ COMPROMISE as defined in the AA Protection Profile
[3] T.KEYING_MATERIAL_ COMPROMISE as defined in the EE Protection Profile
[4] T.AUTHORIZATION_GUESSING as defined in the AA Protection Profile
[5] T.AUTHORIZATION_GUESSING as defined in the EE Protection Profile

Table 3: Threats	
Threat	Description
	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[6]	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 that bypasses the intended security features and provides them unauthorized access to data.
T.UNAUTHORIZED_UPDATE[7]	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 Organizational Security Policies

There are no organizational security policies addressed by this cPP.

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 PP 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

[6] T.UNAUTHORIZED_UPDATE as defined in the AA Protection Profile.
[7] T.UNAUTHORIZED_UPDATE as defined in the EE Protection Profile

Table 4: Assumptions	
Assumption	Description
	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[8]	Users enable Full Drive Encryption on a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption. 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, un-partitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE pre-authentication software) contain no protected data.
A.INITIAL_DRIVE_STATE[9]	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.
A.TRAINED_USER [10]	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 [11]	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 [12]	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.

[8] A.INITIAL_DRIVE_STATE as defined in the AA Protection Profile
[9] A.INITIAL_DRIVE_STATE as defined in the EE Protection Profile
[10] A.TRAINED_USER as defined in the AA Protection Profile
[11] A.TRAINED_USER as defined in the EE Protection Profile
[12] A.POWER_DOWN as defined in the AA Protection Profile

Table 4: Assumptions	
Assumption	Description
	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 [13]	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.

[13] A.POWER_DOWN 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[14]	Volatile memory is cleared after power-off so memory remnant attacks are infeasible.
OE.POWER_DOWN[15]	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.

[14] OE.POWER_DOWN as defined in the AA Protection Profile
[15] OE.POWER_DOWN as defined in the EE Protection Profile

5. Extended Components Definition

This section provides 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

There are no extended Security Functional Requirements defined in this Security Target. All extended SFRs were taken from the cPP.

5.2 Extended Security Assurance Requirements Definitions

There are no extended Security Assurance Requirements defined in this Security Target. All extended SARs were taken from the cPP.

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, and CC Part 3 extended.

6.1 Security Functional Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the cPP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

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)
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)
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)
12	FCS_COP.1(b)	Cryptographic Operation (Hash Algorithm)
13	FCS_COP.1(c)	Cryptographic Operation (Message Authentication)
14	FCS_COP.1(d)	Cryptographic Operation (Key Wrapping)
15	FCS_COP.1(f)	Cryptographic Operation (AES Data Encryption/Decryption)
16	FCS_KYC_EXT.1	Key Chaining (Initiator)
17	FCS_KYC_EXT.2	Key Chaining (Recipient)
18	FCS_PCC_EXT.1	Cryptographic Password Construct and Conditioning
19	FCS_RBG_EXT.1	Random Bit Generation
20	FCS_SNI_EXT.1	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
21	FCS_VAL_EXT.1	Validation
22	FDP_DSK_EXT.1	Protection of Data on Disk
23	FMT_MOF.1	Management of Functions Behavior

Table 6: Security Functional Requirements		
#	SFR	Description
24	FMT_SMF.1(1)	Specification of Management Functions
25	FMT_SMF.1(2)	Specification of Management Functions
26	FMT_SMR.1	Security Roles
27	FPT_KYP_EXT.1	Protection of Key and Key Material
28	FPT_PWR_EXT.1	Power Saving States
29	FPT_PWR_EXT.2	Timing of Power Saving States
30	FPT_TUD_EXT.1(1)	Trusted Update
31	FPT_TUD_EXT.1(2)	Trusted Update
32	FPT_TST_EXT.1	TSF Testing

6.1.1 Class FCS: Cryptographic Support

6.1.1.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

FCS_AFA_EXT.1.1

The TSF shall accept the following authorization factors:

- a submask derived from a password authorization factor conditioned as defined in FCS_PCC_EXT.1.

6.1.1.2 FCS_AFA_EXT.2 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) Cryptographic Key Generation (Symmetric Keys)

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 256 bit that meet the following: No Standard.

6.1.1.4 FCS_CKM.1(c) 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

- generate a DEK using the RBG as specified in FCS_RBG_EXT.1
- accept a DEK that is wrapped as specified in FCS_COP.1(d)

and specified cryptographic key sizes 256 bits.

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

FCS_CKM.4.1(a)

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.

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

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.
  - 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:
    - single overwrite consisting of zeros followed by a read-verify
  - 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) Cryptographic Key Destruction (Software TOE, 3rd Party Storage)

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,
    - a new value of a key,
  - removal of power to the memory,

that meets the following: no standard

6.1.1.8 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 keying material when no longer needed.

6.1.1.9 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.

6.1.1.10 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

FCS_CKM_EXT.6.1

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

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

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-521; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes.

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

FCS_COP.1.1(b)

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-512 that meet the following: ISO/IEC 10118-3:2004.

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

FCS_COP.1.1(c)

The TSF shall perform cryptographic keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-512 and cryptographic key sizes **512 bits** that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

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

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) Cryptographic Operation (AES Data Encryption/Decryption)

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, XTS as specified in IEEE 1619.

6.1.1.16 FCS_KYC_EXT.1 Key Chaining (Initiator)

FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of:

- one, using a submask as the BEV;
- intermediate keys originating from one or more submask(s) to the BEV using the following method(s):
    - 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_KYC_EXT.1.2

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

- without validation taking place.

6.1.1.17 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):

- 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.

6.1.1.18 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

FCS_PCC_EXT.1.1

A password used 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-512, with **1063** iterations, and output cryptographic key sizes 256 bits that meet the following: NIST SP 800-132.

6.1.1.19 FCS_RBG_EXT.1 Extended: 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 Hash_DRBG (any).

FCS_RBG_EXT.1.2

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

- **4** 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.

6.1.1.20 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.

6.1.1.21 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 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 configurable number of consecutive failed validation attempts.

6.1.2 Class FDP: User Data Protection

6.1.2.1 FDP_DSK_EXT.1 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 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 users.

6.1.3.2 FMT_SMF.1(1) Specification of Management Functions

FMT_SMF.1.1(1)

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) define the allowable power saving states,
f) configure the number of failed validation attempts required to trigger corrective behavior.

6.1.3.3 FMT_SMF.1(2) Specification of Management Functions

FMT_SMF.1.1(2)

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) import a wrapped DEK
e) **configure the failed authentication limit count referred to in FCS_SMV_EXT.1.2**
f) **Configure the operational mode of the module**
g) **Change the password conditioned by the PBKDF to unwrap the DEK(in mode 1)**
h) **Change the password conditioned by the PBKDF to unwrap the Black KEK (in mode 6)**.

6.1.3.4 FMT_SMR.1 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_KYP_EXT.1 Extended: Protection of Key and Key Material

FPT_KYP_EXT.1.1(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)
- only store plaintext keys that meet any one of following criteria:
  - The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1.

FPT_KYP_EXT.1.1(2)

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 following criteria:
  - The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.2.

6.1.4.2 FPT_PWR_EXT.1 Power Saving States

FPT_PWR_EXT.1.1 (1)

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

FPT_PWR_EXT.1.1 (2)

The TSF shall define the following Compliant power saving states: D3.

6.1.4.3 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, shutdown, no other conditions.

6.1.4.4 FPT_TUD_EXT.1(1) Trusted Update

FPT_TUD_EXT.1.1(1)

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

FPT_TUD_EXT.1.2(1)

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

FPT_TUD_EXT.1.3(1)

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

6.1.4.5 FPT_TUD_EXT.1(2) Trusted Update

FPT_TUD_EXT.1.1(2)

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

FPT_TUD_EXT.1.2(2)

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

FPT_TUD_EXT.1.3(2)

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.

6.1.4.6 FPT_TST_EXT.1(1) Extended: TSF Testing

FPT_TST_EXT.1.1(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:

- **General module hardware and firmware**
- **AES-256 XTS Encrypt**
- **AES-256 XTS Decrypt**
- **SHA-2**
- **HMAC**
- **AES Key Wrap**
- **PBKDF**
- **DRBG**
- **ECDSA**
- **Temperature & Power Supplies**

6.1.4.7 FPT_TST_EXT.1(2) Extended: TSF Testing

FPT_TST_EXT.1.1(2)

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:

- **NDRBG**
- **DRBG**
- **DRBG Health Check**
- **Module Integrity**

6.2 Security Assurance Requirements

This Security Target is conformant with the assurance requirements specified in the cPPs and Supporting Documents.

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)

6.2.1 Extended Security Assurance Requirements

6.2.1.1 ASE: Security Target

The refined assurance requirement below contains a selection operation mandated by the cPP that was performed by the ST Author.

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.

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 Security Functions:

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

7.1 Cryptographic Support

The TOE supports only one authorization factor, a user-supplied password composed of up to 64 characters with no restrictions. Administrators and Crypto Officers must enforce password requirements to ensure suitable security strength. Passwords may contain upper case letters, lower case letters, numbers, or any other 8-bit value. This password is used in the PBKDF function to produce the KEK that protects the BEV. When entering a Compliant power saving state, the user-supplied password is the only authorization factor used to gain access to user data.

FCS_AFA_EXT.1, FCS_AFA_EXT.2, FCS_PCC_EXT.1

7.1.1 Cryptographic Key Generation and Derivation

Internally generated keys, salts and nonces come from the internal RNG. The generator function uses the Hash_DRBG mechanism described in section 10.1.1 of SP 800-90A. The Hash_DRBG generates a 512-bit pseudorandom number from an 888-bit seed. The 888-bit seed is generated from 4096 Bytes of entropy material using the Hash_df described in section 10.4.1 of SP 800-90A. The 4096 Bytes of entropy material is health checked using methods described in section 4 of (Second DRAFT) NIST Special Publication 800-90B.

In mode 1, the TSF generates a symmetric DEK using the DRBG and uses a user-supplied password, conditioned by the PBKDF, to derive a symmetric KEK used to wrap that key. In operation, the user provides the password, which is used to unwrap the DEK for use. Wrapping is performed using AES-256 in KW mode. The TOE explicitly checks to ensure that both generated XTS AES key halves are not equal, but are unique. The XTS AES tweak value is set to the logical sector value for each sector being accessed.

In mode 6, the symmetric DEK is provided, wrapped using AES-256 in KW mode, by the end user. The TOE explicitly checks to ensure that both received XTS AES key halves are not equal, but are unique. The XTS AES tweak value is set to the logical sector value for each sector being accessed.

In mode 6, the BEV is a 256-bit symmetric Key Encryption Key. This KEK is used to wrap the DEK and is stored in a protected section of memory after being AES key wrapped itself with a 256-bit key generated by the output of the PBKDF (the authorization factor). The administrator provides a plaintext value to become the BEV during initial configuration. This value is encrypted using AES-256 in KW mode, using a user provided password, and is stored in non-volatile memory as the “Black KEK”. During operation in both modes 1 and 6, the BEV is validated as the Black DEK is unwrapped using the derived key.

In mode 1, a user-supplied password is processed by the SP 800-132 compliant PBKDF algorithm to become the BEV. In operational mode 6, the user-supplied password is processed by the SP 800-132 compliant PBKDF algorithm to produce an intermediate key. This key is used to wrap/unwrap the BEV while maintaining an effective minimum strength of 256 bits. In either mode 1 or mode 6, the BEV is 256 bits. (Figure 1)

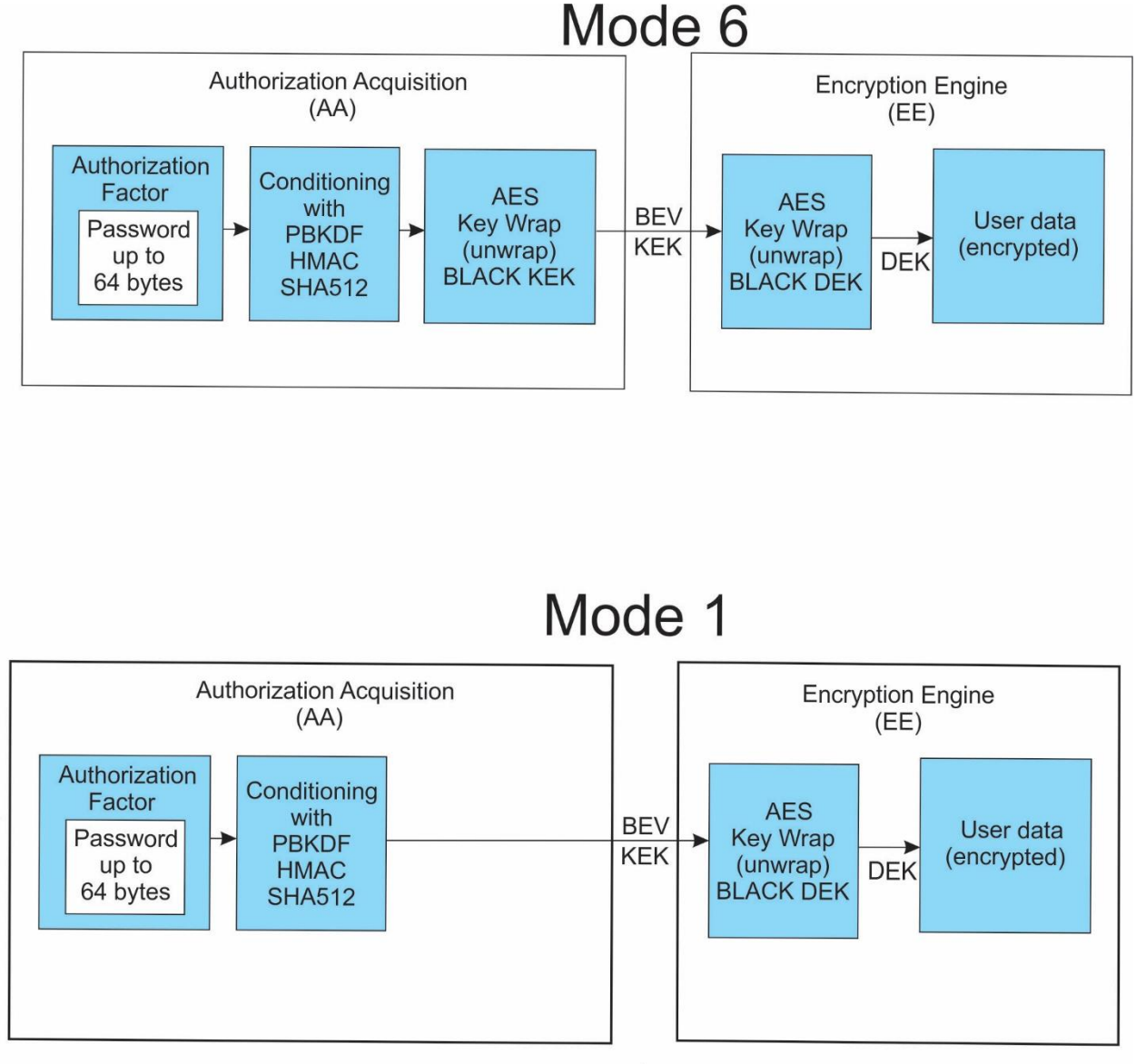

Figure 1 TOE Key Chaining Diagrams

The TSF accepts passwords that are up to 64 characters in length; each character may be any 8-bit value (such as upper case letters, lower case letters, numbers, special characters, ASCII codes, etc.). The supplied password is processed by an SP 800-132 compliant PBKDF using HMAC-SHA-512 over 1063 iterations. The output key is 256 bits.

The TSF contains a password buffer that will accept no more than 64 characters. When administering the TOE using the vendor-supplied utility, the password field is restricted to 64 characters; when administering the TOE without the utility over the ATA interface, the TSF ignores

all characters after the 64th byte. Since the TSF can accept any 8-bit value as a character, no characters are restricted.

FCS_CKM.1, FCS_KYC_EXT.1, FCS_KYC_EXT.2, , FCS_PCC_EXT.1, FCS_VAL_EXT.1, FCS_SNI_EXT.1, FCS_RBG_EXT.1

7.1.2 Cryptographic Key and Key Material Destruction

All plaintext keys and key material are erased when changing modes, loss of power, or when an erase operation is initiated, either automatically when exceeding a crypto officer configured number of sequential failed authentication attempts, or by an explicit initiation of the sanitize operation. Keys are considered no longer needed once the DEK is successfully unwrapped and data is flowing through the encryption engine. Interim keys, such as the Black key and others are erased after they are used even though they are only stored in the TOE's volatile memory. The plaintext KEK used to protect the BEV is kept in the TOE's volatile memory so that a user may change the password. Keys and key material in volatile memory are overwritten by zeros followed by a read-verify. Keys and key material stored in the TOE's Magnetic RAM(MRAM) non-volatile storage are overwritten by a zeroization. The TOE uses the ATA command set to interact with the host platform, however all keys and key material are stored only on the TOE's internal hardware. The TOE never uses the NAND storage media to hold encryption keys, passwords, or any intermediate security related variables. Instead, the plaintext DEK, when unwrapped, resides in the TOE's NAND controller's RAM and wrapped keys reside in a separate NVRAM. The TOE volatile memory is a non-cacheable dual-port FPGA block RAM. The memory controller accesses the TOE volatile memory in 32-bit dwords. The TOE non-volatile memory is a Magnetic RAM (MRAM) device external to the FPGA. The memory controller accesses the TOE non-volatile memory via a custom FPGA implementation that accesses the memory in 32-bit dwords. The drive does not maintain any copies of encryption keys. The plaintext KEK used to protect the BEV along with interim keys, like the Black key, and other key material stored in volatile memory are erased during transition to a Compliant power saving state. As the TOE destroys keys on loss of power, there are no states or circumstances that the TOE does not conform to the key destruction requirements.

Table 8: Cryptographic Key Table – Mode 1

Key	Length (Bits)	Initialization	Usage	Storage	Destruction
BEV	256	Output of PBKDF	Unwrap of Data Encryption Key (DEK)	TOE Volatile Memory	Zeroization
Black DEK (Wrapped DEK)	256	TOE configuration	Protected, wrapped DEK	TOE MRAM Non-volatile memory	Replaced by a new key.
DEK	256	Key Unwrap	Data encryption/decryption	TOE Volatile Memory	Zeroization

Table 9: Cryptographic Key Table – Mode 6					
Key	Length (Bits)	Initialization	Usage	Storage	Destruction
PBKDF Output	256	Authorization Factor Acquisition	Unwrap the Border Encryption Value (BEV)	TOE Volatile Memory	Zeroization
Black KEK (Wrapped BEV)	256	TOE configuration	Protected, wrapped BEV	TOE MRAM Non-volatile memory	Zeroization
BEV	256	Key Unwrap	Unwrap the Data Encryption Key (DEK)	TOE Volatile Memory	Zeroization
Black DEK (Wrapped DEK)	256	Provided at Power On	Protected, wrapped DEK	TOE Volatile memory	Zeroization
DEK	256	Key Unwrap	Data encryption/decryption	TOE Volatile Memory	Zeroization

FCS_CKM_EXT.4, FCS_CKM.4, FCS_CKM_EXT.6

7.1.3 Cryptographic Operations

The module performs the following cryptographic operations. The cryptographic operations take place on an Altera NIOS II which is a CPU and FPGA. The programmed FPGA is referred to as the Armor Processor.

Table 10: Cryptographic Operations		
SFR	Algorithm	Description
FCS_COP.1(f)	XTS-AES-256	The user data is symmetrically encrypted using XTS-AES with a 512 bit key.
FCS_COP.1(d)	AES Key Wrap	All AES-KW operations use a 256 bit key and KW mode as specified in ISO/IEC 18033-3 NIST SP 800-38F. The TOE assumes that the black key entered is wrapped using AES-KW and the TOE itself wraps and unwraps keys (self-generated DEK and black key mode key encryption key).
FCS_COP.1(b)	SHA-512	The module implements SHA-512 with a block size of 1024 which is used in the DRNG as well as used as the hashing function in the HMAC portion of the PBKDF and ECDSA signature verification.
FCS_COP.1(c)	HMAC-SHA-512	Used in SP800-132 PBKDF: 32 byte key, SHA-512 hash, 128-bit block, 512-bit MAC.
FCS_COP.1(a)	ECDSA	Using P-521 and SHA-512 the TOE performs Signature Verification to validate new firmware.
FCS_RBG_EXT.1	RBG	SP 800-90A HASH_DRBG using SHA-512.

The TOE receives the signature data via the ATA DOWNLOAD MICROCODE command. The signature is embedded into the firmware file. The signature verification process begins as soon as the TOE starts receiving data and ends immediately after all required data is received. The data used to verify the digital signature is stored in FPGA block RAM. The TOE does not perform any additional processing that is not part of the digital signature algorithm.

FCS_COP.1, FCS_RBG_EXT.1

7.2 User Data Protection

7.2.1 Protection of Data on Disk

The TOE is initiated by selecting the mode of operation (mode 1 or mode 6), generating or assigning the DEK, entering a user password which gets conditioned by the PBKDF to create a derived key, which depending on the mode will either serve as the BEV, or an intermediate key. This key is used to wrap the DEK in mode 1 and wrapping of the intermediate key in mode 6. The TOE overwrites the password and derived key and leaves only the wrapped DEK stored in NVRAM. The admin (Crypto Officer) exits the role and the TOE is ready to accept data in a user role. At this point, all data on the drive is stored encrypted. All data write operations are performed using the SATA interface and are routed through the encryption engine. There are no SATA data areas that are not encrypted. The TOE has no capability to export key values. The customer is encouraged to use mode 6, where the wrapped DEK is filled externally, so that an unpowered module will contain no CSP information that could be used to unencrypt the data.

The optional write protect port provides a hardware signal that when driven low, will not allow any data on the drive to change or firmware to be updated. The LEDs provide status output only and are used for diagnostic purposes and are not essential to TOE operation.

When writing data to the disk, a host system interfaces with the TOE through a serial interface on the SATA connector Signal Segment. SATA Signals form a bi-directional interface that implements the industry standard Serial ATA (SATA) protocol.

Referring to the diagram below, the SATA Serial-to-Parallel-Conversion logic and the MPU block separate host Control and Status information from the Plaintext data from according the ATA and SATA specifications. This is done by VHDL code (hardware). The Plaintext data from the host flows into the Encryption Logic and is encrypted per the AES-256 XTS specification.

The data must pass through the Encryption core to be stored in NAND so all data becomes encrypted.

The data flow for write operations is:

1. Host to TOE using a SATA serial link,
2. Serial to Parallel conversion,
3. Strip off command data,
4. Plaintext to Encryption logic,
5. cipher text out of Encryption logic to channel logic,
6. then cipher text from channel logic to NAND devices for storage

Read operations are the reverse.

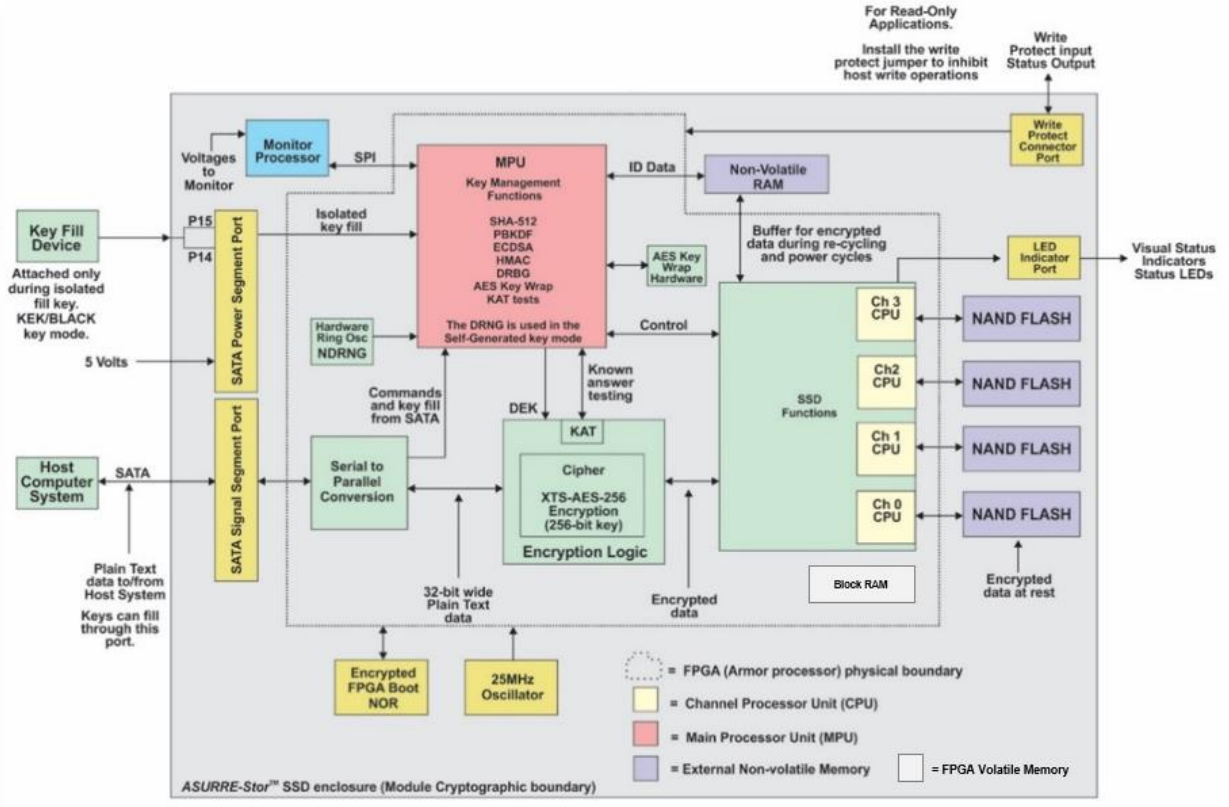

Figure 2 - Asurre-Stor logical flow

FDP_DSK_EXT.1

7.3 Security Management

7.3.1 Specification of Management Functions

The TSF allows the User to change the DEK by cryptographically erasing the DEK and re-provisioning the TOE. The TSF allows the User to cryptographically erase the DEK by issuing the zeroize command or failing an administrator-configurable number of consecutive authentication attempts. The value can be 5, 10, or 15 and the counter of failed authentication attempts persists across power state changes. The TSF allows the User to change the operational mode of the TOE (Mode 1 or Mode 6) or change the password conditioned by the PBKDF to unwrap the DEK (mode 1) or Black KEK (mode 6).

Configuration of the TOE including the number of failed authentication attempts, the operational mode, the configuration password, external Secure Erase trigger input options, etc. is detailed in the Administrative Guidance and is done via the ATA SMART WRITE LOG or ATA WRITE LOG EXT command or preferably through the vendor supplied MDU[16] Utility application which provides a user friendly interface for issuing the command.

When changing the passwords conditioned by the PBKDF to unwrap the DEK or Black KEK, the user provides the current password conditioned by the PBKDF to unwrap the key. The plaintext key is kept in volatile memory; if power interruption occurs, the plaintext key will be lost. The user

[16] The use of the MDU is optional, but was not evaluated as part of the Common Criteria certification process.

provides a new password, which is used to derive the key which is used to wrap the plaintext key. The re-wrapped key is then saved to non-volatile memory, while the plaintext key is kept in volatile memory for use.

The TSF allows the User to initiate the update of the TOE firmware via a series of SATA "DOWNLOAD MICROCODE" commands, but will only accept signed code images as described in Section 7.4.3, and will need to contact Mercury to receive the signed firmware file.

FMT_SMF.1

7.3.2 Security Roles

The TOE supports role-based authentication for a Crypto Officer and a single user. The Crypto Officer role is the role used during the initial secure configuration of the TOE. The Crypto Officer role is authenticated using a Configuration Password. For simplicity of initial configuration, Mercury Systems ships ASURRE-Stor® SSDs with no Configuration Password. The Crypto Officer is responsible for installing the initial Configuration Password during the initial secure configuration procedure. The TOE enters a fully functional User Role from power-on only after completing a successful authentication. Prior to authentication, the User Role cannot write data or read previously stored data.

FMT_SMR.1

7.4 Protection of the TSF

7.4.1 Protection of Key and Key Material

In all modes, the TSF does not store plaintext keys in non-volatile memory. During initial configuration, the plaintext BEV is stored protected by AES-256 in KW mode using a pseudorandom number as the password. This temporary password is replaced after the Administrator configures the password the first time.

FPT_KYP_EXT.1

7.4.2 Power Saving States

In the Evaluated Configuration, the TOE supports the following Compliant power saving state: D3. These Compliant power saving states are configured by administrators when putting the TOE in the Evaluated Configuration. The TOE enters these Compliant power saving states upon request from an authorized user and system shutdown.

FPT_PWR_EXT.1, FPT_PWR_EXT.2, FMT_MOF.1

7.4.3 Trusted Update

The vendor maintains control of the ECDSA P-521 private key which it uses to sign valid firmware updates. The public key is stored as part of the firmware which is integrity checked on power up. When the User initiates a firmware update, the host sends the firmware update to the TSF and the TSF checks the signature. If the signature is valid, the TSF installs the image and reboots to invoke the image. If the signature is not valid, the TSF deletes the image and returns an error.

The TSF also provides the User with a Show Status command which return the HW version and firmware version.

FPT_TUD_EXT.1

7.4.4 TSF Testing

During power up the TSF performs self-tests using known data provided via NIST test vectors. The tests complete in less than 2 seconds. The TSF data that is used for the Known Answer tests is compliant to FIPS 180-4 NIST Test Vectors for the applicable functions and therefore is appropriate for testing. The TSF status indicates if an error is detected. Self-tests and conditional self-tests are listed below.

Table 11: Self-tests

Test Target	Description	When
General module hardware and firmware	Armor™ Processor CRC (32-bit polynomial CRC IEEE 802 standard), MONITOR Firmware Checksum, KATs for Crypto tests, Power supply voltage measurements, Non-Volatile RAM test, NAND Media test, and temperature limits.	Power-on
AES-256 XTS Encrypt	Performs an encryption KAT	Power-on and on demand
AES-256 XTS Decrypt	Performs an decryption KAT	Power-on and on demand
SHA-2	Performs a SHA-512 KAT	Power-on and on demand
HMAC	Performs HMAC SHA-512 KAT.	Power-on and on demand
AES Key Wrap	Performs an encryption KAT and separate decryption KAT.	Power-on and on demand
PBKDF	Performs a KAT using a known password value and compares for an expected MK value.	Power-on and on demand
DRBG	Performs a HASH DRBG KAT using SHA-512.	Power-on, on demand, and prior to creating self-generated random key value
ECDSA	Performs a signature verification KAT	Power on, on demand, and prior to accepting a firmware update.
Temperature, Power supplies, and firmware monitoring	Constant monitoring of temperature, input and internal supply voltages and out of range firmware variables.	Continuously

Table 12: Conditional self-tests

Test Target	Description
NDRNG (entropy source)	Runs a health check test - a repetition count test and adaptive proportion test - as described in SP800-90B section 4.4.1. Test performed continuous for random values requested by the DRNG.
DRBG	KAT Test performed for new random key value generation.
ECDSA Firmware Update	Prior to accepting a new firmware update, an ECDSA signature verification KAT is performed. The new firmware is accepted only if the KAT passes.
DRBG Health check	A KAT performed conditionally per SP 800-90A Section 11.3.
Module Integrity	EXECUTE DEVICE DIAGNOSTIC and SMART OFF-LINE-IMMEDIATE Armor™ Processor CRC (32-bit polynomial CRC IEEE 802 standard), MONITOR Firmware Checksum, KATs for Crypto tests, Power supply voltage measurements, Non-Volatile RAM test, NAND Media test, and temperature limits.

The module enters a failure state when any error is detected. Depending of the severity of the error, the module may no longer be able to communicate with the attached host system. If the error is not severe, the module will abort all write services and return only 0xFF values for read services.

FPT_TST_EXT.1

8. Terms and Definitions

Table 13: 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].
Black KEK	The "black" KEK is a wrapped BEV. The wrapping is accomplished using AES-256 in KW mode and a user-supplied password.
Black DEK	The "black" DEK is a wrapped DEK. The wrapping is accomplished using AES-256 in KW mode and the BEV.
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.
D0 Power Mode	This state is assumed to be the highest level of power consumption. The device is completely active and responsive, and is expected to remember all relevant context continuously.
D3 Power Mode	Power has been fully removed from the device.
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.
MDU	The MDU utility runs on Windows XP, Windows 7 and Windows 10. MDU lists all the Mercury Systems ASURRE-Stor® SSDs detected within a system. The initial secure configuration of the TOE can be accomplished using the Mercury Systems MDU Utility.
Non-Volatile Memory	A type of computer memory that will retain information without power.
Operating System (OS)	Software which runs at the highest privilege level and can directly control hardware resources.

Table 13: cPP Glossary	
Term	Description
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 14: 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
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
NIST	National Institute of Standards and Technology
OS	Operating System
RBG	Random Bit Generator
RNG	Random Number Generator
RSA	Rivest Shamir Adleman Algorithm
SAR	Security Assurance Requirement

Table 14: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
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
USB	Universal Serial Bus
XOR	Exclusive or
XTS	XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing

9. References

Table 15: TOE Guidance Documentation		
Reference	Description	Date
[1]	asurreStor-AdministrativeGuide	2/4/2020
[2]	SSD Secure Configuration Programmer's Guide	11/21/2019

Table 16: 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 17: Supporting Documentation			
Reference	Description	Version	Date
[7]	collaborative Protection Profile for Full Drive Encryption – Encryption Engine	2.0 + Errata 20190201	February 1, 2019
[8]	Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition	2.0 + Errata 20190201	February 2019
[9]	collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition	2.0 + Errata 20190201	February 1, 2019
[10]	Supporting Document Mandatory Technical Document Full Drive Encryption: Authorization Acquisition	2.0 + Errata 20190201	February 2019