

---

# **Curtiss-Wright Defense Solutions XMC NVMe Encryptor version 1.0.0 Security Target**

Version 0.5  
11/01/2024

---

*Prepared for:*

**Curtiss-Wright Defense Solutions**

2600 Paramount Pl #200  
Fairborn, OH 45324

*Prepared By:*



[www.gossamersec.com](http://www.gossamersec.com)

<b>1. SECURITY TARGET INTRODUCTION .....</b>	<b>3</b>
1.1 SECURITY TARGET REFERENCE.....	3
1.2 TOE REFERENCE.....	3
1.3 TOE OVERVIEW .....	4
1.4 TOE DESCRIPTION .....	4
1.4.1 TOE Architecture.....	4
1.4.2 TOE Documentation.....	5
<b>2. CONFORMANCE CLAIMS.....</b>	<b>6</b>
2.1 CONFORMANCE RATIONALE.....	6
<b>3. SECURITY OBJECTIVES .....</b>	<b>7</b>
3.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT .....	7
<b>4. EXTENDED COMPONENTS DEFINITION .....</b>	<b>8</b>
<b>5. SECURITY REQUIREMENTS.....</b>	<b>9</b>
5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS .....	9
5.1.1 Cryptographic support (FCS).....	10
5.1.2 User data protection (FDP).....	16
5.1.3 Security management (FMT) .....	16
5.1.4 Protection of the TSF (FPT).....	17
5.2 TOE SECURITY ASSURANCE REQUIREMENTS.....	19
5.2.1 Development (ADV).....	19
5.2.2 Guidance documents (AGD).....	20
5.2.3 Life-cycle support (ALC) .....	21
5.2.4 Security Target (ASE).....	21
5.2.5 Tests (ATE).....	22
5.2.6 Vulnerability assessment (AVA).....	22
<b>6. TOE SUMMARY SPECIFICATION.....</b>	<b>23</b>
6.1 CRYPTOGRAPHIC SUPPORT .....	23
6.2 USER DATA PROTECTION .....	25
6.3 SECURITY MANAGEMENT .....	25
6.4 PROTECTION OF THE TSF .....	25
<b>7. KEY MANAGEMENT DESCRIPTION.....</b>	<b>27</b>

## LIST OF TABLES

<b>Table 1 TOE Security Functional Components .....</b>	<b>10</b>
<b>Table 2 Assurance Components .....</b>	<b>19</b>
<b>Table 3 3<sup>rd</sup> Party Hardware Components .....</b>	<b>21</b>
<b>Table 4 Cryptographic Algorithms .....</b>	<b>24</b>
<b>Table 5 Key Identification.....</b>	<b>27</b>

## 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Curtiss-Wright Defense Solutions XMC NVMe Encryptor provided by Curtiss-Wright Defense Solutions. The TOE is being evaluated as a hardware full drive encryption solution.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

### Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement. Operations performed in the Protection Profiles are not marked in the ST. The conventions below are for ST operations exclusively.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP\_ACC.1(1) and FDP\_ACC.1(2) indicate that the ST includes two iterations of the FDP\_ACC.1 requirement.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [*selected-assignment*]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [*selection*]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "... **all** objects ..." or "... ~~big~~ things ...").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

### 1.1 Security Target Reference

**ST Title** – Curtiss-Wright Defense Solutions XMC NVMe Encryptor version 1.0.0 Security Target

**ST Version** – Version 0.5

**ST Date** – 11/01/2024

### 1.2 TOE Reference

**TOE Identification** – Curtiss-Wright Defense Solutions XMC NVMe Encryptor version 1.0.0

**TOE Developer** – Curtiss-Wright Defense Solutions

**Evaluation Sponsor** – Curtiss-Wright Defense Solutions

---

## 1.3 TOE Overview

---

The Target of Evaluation (TOE) is Curtiss-Wright Defense Solutions XMC NVMe Encryptor version 1.0.0.

The TOE provides hardware Full Drive Encryption of standard NVMe compliant drives.

---

## 1.4 TOE Description

---

The Curtiss-Wright Defense Solutions XMC NVMe Encryptor (hereafter referred to as the TOE) is an in-line hardware encryption layer that is used for Data-At-Rest (DAR) encryption of attached NVMe drives and can be installed in any host system that supports installation of the assembly. The TOE contains two external interfaces – one for serial connection and one ethernet port for an HTTPS connection. These connections are used for management.

To be in compliance with NIAP technical decision 606, the TOE may use the serial cable for management or use HTTPS if done on a private, dedicated network.

---

### 1.4.1 TOE Architecture

---

The TOE provides a hardware Full Drive Encryption solution that can encrypt standard NVMe compliant drives.

---

#### 1.4.1.1 Physical Boundaries

---

The TOE's physical boundary is the physical perimeter of its enclosure. The TOE provides a ruggedized solution to secure Data at Rest (DAR).

---

#### 1.4.1.2 Logical Boundaries

---

This section summarizes the security functions provided by NVMe Encryptor (HW Layer):

- Cryptographic support
- User data protection
- Security management
- Protection of the TSF

---

##### 1.4.1.2.1 Cryptographic support

---

The TOE includes cryptographic functionality for key management, user authentication, and block-based encryption including: symmetric key generation, encryption/decryption, cryptographic hashing, keyed-hash message authentication, and password-based key derivation. These functions are supported with suitable random bit generation, key derivation, salt generation, initialization vector generation, secure key storage, and key destruction. These primitive cryptographic functions are used to encrypt Data-At-Rest (including the generation and protection of keys and key encryption keys) used by the TOE.

---

##### 1.4.1.2.2 User data protection

---

The TOE performs Full Drive Encryption on the entire drive (so that no plaintext exists) and does so without user intervention.

---

##### 1.4.1.2.3 Security management

---

The TOE provides each of the required management services necessary to manage the full drive encryption using a command line interface.

---

#### 1.4.1.2.4 Protection of the TSF

---

The TOE implements a number of features to protect itself to ensure the reliability and integrity of its security features. It protects key and key material, and includes functions to perform self-tests and software/firmware integrity checking so that it might detect when it is failing or may be corrupt. If any of the self-tests fails, the TOE will not go into an operational mode.

---

#### 1.4.2 TOE Documentation

---

Curtiss-Wright NVMe Encryptor User Guide, DDOC0182-000-NIAP A2 [User Guide]

## 2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
  - Part 3 Conformant
- Package Claims:
  - collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201, February 1, 2019 and
  - collaborative Protection Profile for Full Drive Encryption Authorization Acquisition, Version 2.0 + Errata 20190201, February 1, 2019 (FDEEEcPP20E/FDEAAcPP20E)
- Technical Decisions:

TD No.	PP	Applies?	Rationale
TD0769	FDEEEcPP20E/ FDEAAcPP20E	Yes	SFR is claimed
TD0767	FDEAAcPP20E	Yes	SFR is claimed
TD0766	FDEEEcPP20E/ FDEAAcPP20E	Yes	SFR is claimed
TD0765	FDEAAcPP20E	Yes	SFR is claimed
TD0764	FDEAAcPP20E	Yes	SFR is claimed
TD0760	FDEAAcPP20E	Yes	SFR is claimed
TD0759	FDEAAcPP20E	Yes	SFR is claimed
TD0606	FDEEEcPP20E/ FDEAAcPP20E	Yes	Product is a NAS
TD0464	FDEEEcPP20E	Yes	SFR is claimed
TD0460	FDEEEcPP20E	Yes	SFR is claimed
TD0458	FDEEEcPP20E/FDEAAcPP20E	Yes	SFR is claimed

### 2.1 Conformance Rationale

The ST conforms to the FDEEEcPP20E/FDEAAcPP20E. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

---

### 3. Security Objectives

---

The Security Problem Definition may be found in the FDEEEcPP20E/FDEAAcPP20E and this section reproduces only the corresponding Security Objectives for the operational environment for reader convenience. The FDEEEcPP20E/FDEAAcPP20E offers additional information about the identified security objectives, but that has not been reproduced here and the FDEEEcPP20E/FDEAAcPP20E should be consulted if there is interest in that material.

In general, the FDEEEcPP20E/FDEAAcPP20E has defined Security Objectives appropriate for Full Drive Encryption and as such are applicable to the Curtiss-Wright Defense Solutions XMC NVMe Encryptor TOE.

---

#### 3.1 Security Objectives for the Operational Environment

---

**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.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.

**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.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.POWER\_DOWN** Volatile memory is cleared after power-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.TRUSTED\_CHANNEL** Communication among and between product components (i.e., AA and EE) is sufficiently protected to prevent information disclosure.

## 4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the FDEEEcPP20E/FDEAAcPP20E. The FDEEEcPP20E/FDEAAcPP20E defines the following extended requirements and since they are not redefined in this ST the FDEEEcPP20E/FDEAAcPP20E should be consulted for more information in regard to those CC extensions.

### Extended SFRs:

- FDEAAcPP20E:FCS\_AFA\_EXT.1: Authorization Factor Acquisition
- FDEAAcPP20E:FCS\_AFA\_EXT.2: Timing of Authorization Factor Acquisition
- FDEAAcPP20E:FCS\_CKM\_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
- FDEEEcPP20E:FCS\_CKM\_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
- FDEAAcPP20E:FCS\_CKM\_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
- FDEEEcPP20E:FCS\_CKM\_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
- FDEEEcPP20E:FCS\_CKM\_EXT.6: Cryptographic Key Destruction Types
- FDEAAcPP20E:FCS\_KDF\_EXT.1: Cryptographic Key Derivation
- FDEEEcPP20E:FCS\_KDF\_EXT.1: Cryptographic Key Derivation
- FDEAAcPP20E:FCS\_KYC\_EXT.1: Key Chaining (Initiator)
- FDEEEcPP20E:FCS\_KYC\_EXT.2: Key Chaining (Recipient)
- FDEAAcPP20E:FCS\_PCC\_EXT.1: Cryptographic Password Construct and Conditioning
- FDEAAcPP20E:FCS\_RBG\_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FDEEEcPP20E:FCS\_RBG\_EXT.1: Random Bit Generation
- FDEAAcPP20E:FCS\_SNI\_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- FDEEEcPP20E:FCS\_SNI\_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- FDEAAcPP20E:FCS\_VAL\_EXT.1: Validation
- FDEEEcPP20E:FCS\_VAL\_EXT.1: Validation
- FDEEEcPP20E:FDP\_DSK\_EXT.1: Protection of Data on Disk
- FDEAAcPP20E:FPT\_KYP\_EXT.1: Protection of Key and Key Material
- FDEEEcPP20E:FPT\_KYP\_EXT.1: Protection of Key and Key Material
- FDEAAcPP20E:FPT\_PWR\_EXT.1: Power Saving States
- FDEEEcPP20E:FPT\_PWR\_EXT.1: Power Saving States
- FDEAAcPP20E:FPT\_PWR\_EXT.2: Timing of Power Saving States
- FDEEEcPP20E:FPT\_PWR\_EXT.2: Timing of Power Saving States
- FDEAAcPP20E:FPT\_TST\_EXT.1: TSF Testing
- FDEEEcPP20E:FPT\_TST\_EXT.1: TSF Testing
- FDEAAcPP20E:FPT\_TUD\_EXT.1: Trusted Update
- FDEEEcPP20E:FPT\_TUD\_EXT.1: Trusted Update



## 5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the FDEEEcPP20E/FDEAAcPP20E. The refinements and operations already performed in the FDEEEcPP20E/FDEAAcPP20E are not identified (e.g., highlighted) here, rather the requirements have been copied from the FDEEEcPP20E/FDEAAcPP20E and any residual operations have been completed herein. Of particular note, the FDEEEcPP20E/FDEAAcPP20E made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the FDEEEcPP20E/FDEAAcPP20E. The FDEEEcPP20E/FDEAAcPP20E should be consulted for the assurance activity definitions.

### 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Curtiss-Wright Defense Solutions XMC NVMe Encryptor TOE.

Requirement Class	Requirement Component
<b>FCS: Cryptographic support</b>	FDEAAcPP20E:FCS_AFA_EXT.1: Authorization Factor Acquisition
	FDEAAcPP20E:FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition
	FDEAAcPP20E:FCS_CKM.1(b): Cryptographic Key Generation (Symmetric Keys)
	FDEEEcPP20E:FCS_CKM.1(b): Cryptographic Key Generation (Symmetric Keys)
	FDEEEcPP20E:FCS_CKM.1(c): Cryptographic Key Generation (Data Encryption Key)
	FDEAAcPP20E:FCS_CKM.4(a): Cryptographic Key Destruction (Power Management)
	FDEEEcPP20E:FCS_CKM.4(a): Cryptographic Key Destruction (Power Management)
	FDEEEcPP20E:FCS_CKM.4(b): Cryptographic Key Destruction (TOE Controlled Hardware)
	FDEAAcPP20E:FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
	FDEEEcPP20E:FCS_CKM.4(e): Cryptographic Key Destruction (Key Cryptographic Erase)
	FDEAAcPP20E:FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
	FDEEEcPP20E:FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
	FDEAAcPP20E:FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
	FDEEEcPP20E:FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
	FDEEEcPP20E:FCS_CKM_EXT.6: Cryptographic Key Destruction Types
	FDEAAcPP20E:FCS_COP.1(a): Cryptographic Operation (Signature Verification)
	FDEEEcPP20E:FCS_COP.1(a): Cryptographic Operation (Signature Verification)
	FDEAAcPP20E:FCS_COP.1(b): Cryptographic operation (Hash Algorithm)
	FDEEEcPP20E:FCS_COP.1(b): Cryptographic Operation (Hash Algorithm)
	FDEAAcPP20E:FCS_COP.1(c): Cryptographic operation (Keyed Hash Algorithm)
	FDEEEcPP20E:FCS_COP.1(c): Cryptographic Operation (Message Authentication)
	FDEAAcPP20E:FCS_COP.1(d): Cryptographic operation (Key Wrapping)
	FDEEEcPP20E:FCS_COP.1(d): Cryptographic Operation (Key Wrapping)

	FDEAAcPP20E:FCS_COP.1(f): Cryptographic operation (AES Data Encryption/Decryption)
	FDEEEcPP20E:FCS_COP.1(f): Cryptographic Operation (AES Data Encryption/Decryption)
	FDEAAcPP20E:FCS_COP.1(g): Cryptographic operation (Key Encryption)
	FDEEEcPP20E:FCS_COP.1(g): Cryptographic Operation (Key Encryption)
	FDEAAcPP20E:FCS_KDF_EXT.1: Cryptographic Key Derivation
	FDEEEcPP20E:FCS_KDF_EXT.1: Cryptographic Key Derivation
	FDEAAcPP20E:FCS_KYC_EXT.1: Key Chaining (Initiator)
	FDEEEcPP20E:FCS_KYC_EXT.2: Key Chaining (Recipient)
	FDEAAcPP20E:FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning
	FDEAAcPP20E:FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FDEEEcPP20E:FCS_RBG_EXT.1: Random Bit Generation
	FDEAAcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
	FDEEEcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
	FDEAAcPP20E:FCS_VAL_EXT.1: Validation
	FDEEEcPP20E:FCS_VAL_EXT.1: Validation
<b>FDP: User data protection</b>	FDEEEcPP20E:FDP_DSK_EXT.1: Protection of Data on Disk
<b>FMT: Security management</b>	FDEAAcPP20E:FMT_MOF.1: Management of Functions Behavior
	FDEAAcPP20E:FMT_SMF.1: Specification of Management Functions
	FDEEEcPP20E:FMT_SMF.1: Specification of Management Functions
	FDEAAcPP20E:FMT_SMR.1: Security Roles
<b>FPT: Protection of The TSF</b>	FDEEEcPP20E:FPT_FUA_EXT.1: Firmware Update Authentication
	FDEAAcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material
	FDEEEcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material
	FDEAAcPP20E:FPT_PWR_EXT.1: Power Saving States
	FDEEEcPP20E:FPT_PWR_EXT.1: Power Saving States
	FDEAAcPP20E:FPT_PWR_EXT.2: Timing of Power Saving States
	FDEEEcPP20E:FPT_PWR_EXT.2: Timing of Power Saving States
	FDEAAcPP20E:FPT_TST_EXT.1: TSF Testing
	FDEEEcPP20E:FPT_TST_EXT.1: TSF Testing
	FDEAAcPP20E:FPT_TUD_EXT.1: Trusted Update
	FDEEEcPP20E:FPT_TUD_EXT.1: Trusted Update

Table 1 TOE Security Functional Components

### 5.1.1 Cryptographic support (FCS)

#### 5.1.1.1 Authorization Factor Acquisition (FDEAAcPP20E:FCS\_AFA\_EXT.1)

##### FDEAAcPP20E: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]*. (TD0759 applied)

### 5.1.1.2 Timing of Authorization Factor Acquisition (FDEAAcPP20E:FCS\_AFA\_EXT.2)

#### FDEAAcPP20E: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.

### 5.1.1.3 Cryptographic key generation (Symmetric Keys) (FDEAAcPP20E:FCS\_CKM.1(b))

#### FDEAAcPP20E: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.

### 5.1.1.4 Cryptographic Key Generation (Symmetric Keys) (FDEEEcPP20E:FCS\_CKM.1(b))

#### FDEEEcPP20E: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.

### 5.1.1.5 Cryptographic Key Generation (Data Encryption Key) (FDEEEcPP20E:FCS\_CKM.1(c))

#### FDEEEcPP20E:FCS\_CKM.1.1(c)

Refinement: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation method [  
- *accept a DEK that is wrapped as specified in FCS\_COP.1(d)*] and specified cryptographic key sizes [**256 bits**].

### 5.1.1.6 Cryptographic Key Destruction (Power Management) (FDEAAcPP20E:FCS\_CKM.4(a))

#### FDEAAcPP20E: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.4(d).

### 5.1.1.7 Cryptographic Key Destruction (Power Management) (FDEEEcPP20E:FCS\_CKM.4(a))

#### FDEEEcPP20E: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.

### 5.1.1.8 Cryptographic Key Destruction (TOE-Controlled Hardware) (FDEEEcPP20:FCS\_CKM.4(b))

#### FDEEEcPP20: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* [  
o [*single overwrite consisting of* [  
▪ *zeros*  
▪ *ones*]  
o - *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* [  
o [*single*] *overwrite consisting of zeros followed by a read-verify,*  
o [*single*] *overwrite consisting of ones followed by a read-verify,*

- *overwrite with a new value of a key of the same size followed by a read-verify,*
  - *[single] overwrite consisting of [data from an RBG] followed by a read-verify*  
*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.

#### 5.1.1.9 Cryptographic Key Destruction (Software TOE, 3<sup>rd</sup> Party Storage) (FDEAAcPP20:FCS\_CKM.4(d))

##### FDEAAcPP20:FCS\_CKM.4.1(d)

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 overwright consisting of [*
    - *zeros]*
    - *- removal of power to the memory] ]*
- *For non-volatile storage that consists of the invocation of an interface provided by the underlying platform that [logically addresses the storage location of the key and performs a*
  - *[single] overwrite consisting of [*
    - *zeros*
    - *ones]]]*

] that meets the following: no standard.

#### 5.1.1.10 Cryptographic Key Destruction (Key Cryptographic Erase) (FDEEEcPP20E:FCS\_CKM.4(e))

##### FDEEEcPP20E:FCS\_CKM.4.1(e)

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method by using the appropriate method to destroy all encryption keys encrypting the key intended for destruction that meets the following: no standard.

#### 5.1.1.11 Cryptographic Key and Key Material Destruction (Destruction Timing) (FDEAAcPP20E:FCS\_CKM\_EXT.4(a))

##### FDEAAcPP20E:FCS\_CKM\_EXT.4.1(a)

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

#### 5.1.1.12 Cryptographic Key and Key Material Destruction (Destruction Timing) (FDEEEcPP20E:FCS\_CKM\_EXT.4(a))

##### FDEEEcPP20E:FCS\_CKM\_EXT.4.1(a)

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

#### 5.1.1.13 Cryptographic Key and Key Material Destruction (Power Management) (FDEAAcPP20E:FCS\_CKM\_EXT.4(b))

##### FDEAAcPP20E:FCS\_CKM\_EXT.4.1(b)

Refinement: 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.

#### 5.1.1.14 Cryptographic Key and Key Material Destruction (Power Management) (FDEEEcPP20E:FCS\_CKM\_EXT.4(b))

##### FDEEEcPP20E: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.

---

**5.1.1.15 Cryptographic Key Destruction Types (FDEEEcPP20E:FCS\_CKM\_EXT.6)**

---

**FDEEEcPP20E:FCS\_CKM\_EXT.6.1**

The TSF shall use [*FCS\_CKM.4(b)*] key destruction methods.

---

**5.1.1.16 Cryptographic Operation (Signature Verification) (FDEAAcPP20E:FCS\_COP.1(a))**

---

**FDEAAcPP20E:FCS\_COP.1.1(a)**

Refinement: 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 meet the following: [

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

---

**5.1.1.17 Cryptographic Operation (Signature Verification) (FDEEEcPP20E:FCS\_COP.1(a))**

---

**FDEEEcPP20E:FCS\_COP.1.1(a)**

Refinement: 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 meet the following: [

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

---

**5.1.1.18 Cryptographic operation (Hash Algorithm) (FDEAAcPP20E:FCS\_COP.1(b))**

---

**FDEAAcPP20E:FCS\_COP.1.1(b)**

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

---

**5.1.1.19 Cryptographic Operation (Hash Algorithm) (FDEEEcPP20E:FCS\_COP.1(b))**

---

**FDEEEcPP20E:FCS\_COP.1.1(b)**

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

---

**5.1.1.20 Cryptographic operation (Keyed Hash Algorithm) (FDEAAcPP20E:FCS\_COP.1(c))**

---

**FDEAAcPP20E:FCS\_COP.1.1(c)**

Refinement: The TSF shall perform cryptographic keyed-hash message authentication in accordance with a specified cryptographic algorithm [*HMAC-SHA-256, HMAC-SHA-384*] and cryptographic key sizes [*256, 384*] that meet the following: [*ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'*].

---

**5.1.1.21 Cryptographic Operation (Message Authentication) (FDEEEcPP20E:FCS\_COP.1(c))**

---

**FDEEEcPP20E:FCS\_COP.1.1(c)**

Refinement: The TSF shall perform [message authentication] in accordance with a specified cryptographic algorithm [*HMAC-SHA-256, HMAC-SHA-384*] and cryptographic key sizes [*256 and 384-bit keys used in [HMAC]*] that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

---

**5.1.1.22 Cryptographic operation (Key Wrapping) (FDEAAcPP20E:FCS\_COP.1(d))**

---

**FDEAAcPP20E: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*].

---

#### 5.1.1.23 Cryptographic Operation (Key Wrapping) (FDEEEcPP20E:FCS\_COP.1(d))

---

##### FDEEEcPP20E: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*].

---

#### 5.1.1.24 Cryptographic operation (AES Data Encryption/Decryption) (FDEAAcPP20E:FCS\_COP.1(f))

---

##### FDEAAcPP20E: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 /IEC 18033-3, [*XTS as specified in IEEE 1619*].

---

#### 5.1.1.25 Cryptographic Operation (AES Data Encryption/Decryption) (FDEEEcPP20E:FCS\_COP.1(f))

---

##### FDEEEcPP20E: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*].

---

#### 5.1.1.26 Cryptographic Key Derivation (FDEAAcPP20E:FCS\_KDF\_EXT.1)

---

##### FDEAAcPP20E: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.

---

#### 5.1.1.27 Cryptographic Key Derivation (FDEEEcPP20E:FCS\_KDF\_EXT.1)

---

##### FDEEEcPP20E: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.

---

#### 5.1.1.28 Key Chaining (Initiator) (FDEAAcPP20E:FCS\_KYC\_EXT.1)

---

##### FDEAAcPP20E:FCS\_KYC\_EXT.1.1

The TSF shall maintain a key chain of: [*intermediate keys originating from one or more submask(s) to the BEV using the following method(s): [key derivation as specified in FCS\_KDF\_EXT.1]*] while maintaining an effective strength of [*256 bits*] for symmetric keys and an effective strength of [*not applicable*] for asymmetric keys.

##### FDEAAcPP20E:FCS\_KYC\_EXT.1.2

The TSF shall provide at least a [*256 bit*] BEV to [*the encryption engine*] [- *without validation taking place*].

---

#### 5.1.1.29 Key Chaining (Recipient) (FDEEEcPP20E:FCS\_KYC\_EXT.2)

---

##### FDEEEcPP20E:FCS\_KYC\_EXT.2.1

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

##### FDEEEcPP20E: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] for symmetric keys and an effective strength of [not applicable] for asymmetric keys.

#### 5.1.1.30 Cryptographic Password Construct and Conditioning (FDEAAcPP20E:FCS\_PCC\_EXT.1)

##### FDEAAcPP20E:FCS\_PCC\_EXT.1.1

A password used by the TSF to generate a password authorization factor shall enable up to [64] characters in the set of upper case characters, lower case characters, numbers, and [no special characters] and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[SHA-256], with [1000] iterations, and output cryptographic key sizes [256 bits] that meet the following: NIST SP 800-132. (TD0764 applied)

#### 5.1.1.31 Extended: Cryptographic Operation (Random Bit Generation) (FDEAAcPP20E:FCS\_RBG\_EXT.1)

##### FDEAAcPP20E: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)].

##### FDEAAcPP20E:FCS\_RBG\_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [one] hardware-based noise source(s) with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

#### 5.1.1.32 Random Bit Generation (FDEEEcPP20E:FCS\_RBG\_EXT.1)

##### FDEEEcPP20E: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)].

##### FDEEEcPP20E:FCS\_RBG\_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [one] hardware-based noise source(s) with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

#### 5.1.1.33 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FDEAAcPP20E:FCS\_SNI\_EXT.1)

##### FDEAAcPP20E:FCS\_SNI\_EXT.1.1

The TSF shall [use salts that are generated by a [DRBG as specified in FCS\_RBG\_EXT.1]].

##### FDEAAcPP20E:FCS\_SNI\_EXT.1.2

The TSF shall use [no nonces].

##### FDEEEcPP20E: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]. (TD0760 applied)

#### 5.1.1.34 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FDEEEcPP20E:FCS\_SNI\_EXT.1)

##### FDEEEcPP20E:FCS\_SNI\_EXT.1.1

The TSF shall [use salts that are generated by a [DRBG as specified in FCS\_RBG\_EXT.1]].

##### FDEEEcPP20E:FCS\_SNI\_EXT.1.2

The TSF shall use [no nonces].

##### FDEEEcPP20E: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]. (TD0760 applied)

---

**5.1.1.35 Validation (FDEAAcPP20E:FCS\_VAL\_EXT.1)**

---

**FDEAAcPP20E: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)*]

**FDEAAcPP20E:FCS\_VAL\_EXT.1.2**

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

**FDEAAcPP20E:FCS\_VAL\_EXT.1.3**

The TSF shall [- *block validation after [five] consecutive failed validation attempts*].

---

**5.1.1.36 Validation (FDEEEcPP20E:FCS\_VAL\_EXT.1)**

---

**FDEEEcPP20E: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)*]

**FDEEEcPP20E: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.

**FDEEEcPP20E:FCS\_VAL\_EXT.1.3**

The TSF shall [- *block validation after [five] consecutive failed validation attempts and require a reboot before any more attempts*].

---

**5.1.2 User data protection (FDP)**

---

**5.1.2.1 Protection of Data on Disk (FDEEEcPP20E:FDP\_DSK\_EXT.1)**

---

**FDEEEcPP20E: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.

**FDEEEcPP20E:FDP\_DSK\_EXT.1.2**

The TSF shall encrypt all protected data without user intervention.

---

**5.1.3 Security management (FMT)**

---

**5.1.3.1 Management of Functions Behavior (FDEAAcPP20E:FMT\_MOF.1)**

---

**FDEAAcPP20E: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].

---

**5.1.3.2 Specification of Management Functions (FDEAAcPP20E:FMT\_SMF.1)**

---

**FDEAAcPP20E:FMT\_SMF.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 values or set of authorization values used within the supported authorization method,
- d) initiate TOE firmware/software updates,
- e) [[*import a wrapped DEK*]] (TD0767 applied)



### 5.1.3.3 Specification of Management Functions (FDEEEcPP20E:FMT\_SMF.1)

#### FDEEEcPP20E:FMT\_SMF.1.1

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

- a) change the DEK, as specified in FCS\_CKM.1, when reprovisioning 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][*.

### 5.1.3.4 Security Roles (FDEAAcPP20E:FMT\_SMR.1)

#### FDEAAcPP20E:FMT\_SMR.1.1

The TSF shall maintain the roles [authorized user].

#### FDEAAcPP20E:FMT\_SMR.1.2

The TSF shall be able to associate users with roles.

## 5.1.4 Protection of the TSF (FPT)

### 5.1.4.1 Firmware Update Authentication (FDEEEcPP20E:FPT\_FUA\_EXT.1)

#### FDEEEcPP20E: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 *[the public key]*.

#### FDEEEcPP20E: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).

#### FDEEEcPP20E: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.

#### FDEEEcPP20E:FPT\_FUA\_EXT.1.4

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

### 5.1.4.2 Protection of Key and Key Material (FDEAAcPP20E:FPT\_KYP\_EXT.1)

#### FDEAAcPP20E: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),*
- *only store plaintext keys that meet any one of the following criteria [*
  - *The plaintext key is not part of the key chain as specified in FCS\_KYC\_EXT.1]*

]. (TD0769 applied)

### 5.1.4.3 Protection of Key and Key Material (FDEEEcPP20E:FPT\_KYP\_EXT.1)

#### FDEEEcPP20E: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),*
- *only store plaintext keys that meet any one of the following criteria [*
  - *The plaintext key is not part of the key chain as specified in FCS\_KYC\_EXT.2]*

]. (TD0769 applied)

---

#### 5.1.4.4 Power Saving States (FDEAAcPP20E:FPT\_PWR\_EXT.1)

---

##### FDEAAcPP20E:FPT\_PWR\_EXT.1.1

The TSF shall define the following Compliant power saving states: [G3].

---

#### 5.1.4.5 Power Saving States (FDEEEcPP20E:FPT\_PWR\_EXT.1)

---

##### FDEEEcPP20E:FPT\_PWR\_EXT.1.1

The TSF shall define the following Compliant power saving states: [G3].  
(TD0464 applied)

---

#### 5.1.4.6 Timing of Power Saving States (FDEAAcPP20E:FPT\_PWR\_EXT.2)

---

##### FDEAAcPP20E: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].

---

#### 5.1.4.7 Timing of Power Saving States (FDEEEcPP20E:FPT\_PWR\_EXT.2)

---

##### FDEEEcPP20E: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].

---

#### 5.1.4.8 TSF Testing (FDEAAcPP20E:FPT\_TST\_EXT.1)

---

##### FDEAAcPP20E: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: [*Cryptographic Algorithm Self-tests*].

---

#### 5.1.4.9 TSF Testing (FDEEEcPP20E:FPT\_TST\_EXT.1)

---

##### FDEEEcPP20E: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: [*Cryptographic Algorithm Self-tests*].

---

#### 5.1.4.10 Trusted Update (FDEAAcPP20E:FPT\_TUD\_EXT.1)

---

##### FDEAAcPP20E:FPT\_TUD\_EXT.1.1

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

##### FDEAAcPP20E:FPT\_TUD\_EXT.1.2

Refinement: The TSF shall provide authorized users the ability to initiate updates to TOE [*firmware*].

##### FDEAAcPP20E:FPT\_TUD\_EXT.1.3

Refinement: 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.

### 5.1.4.11 Trusted Update (FDEEEcPP20E:FPT\_TUD\_EXT.1)

#### FDEEEcPP20E:FPT\_TUD\_EXT.1.1

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

#### FDEEEcPP20E:FPT\_TUD\_EXT.1.2

Refinement: The TSF shall provide authorized users the ability to initiate updates to TOE [*firmware*].

#### FDEEEcPP20E: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.

## 5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
<b>ADV: Development</b>	ADV FSP.1: Basic functional specification
<b>AGD: Guidance documents</b>	AGD OPE.1: Operational user guidance
	AGD PRE.1: Preparative procedures
<b>ALC: Life-cycle support</b>	ALC CMC.1: Labelling of the TOE
	ALC CMS.1: TOE CM coverage
<b>ATE: Tests</b>	ATE IND.1: Independent testing - conformance
<b>AVA: Vulnerability assessment</b>	AVA VAN.1: Vulnerability survey

Table 2 Assurance Components

### 5.2.1 Development (ADV)

#### 5.2.1.1 Basic functional specification (ADV\_FSP.1)

##### ADV\_FSP.1.1d

The developer shall provide a functional specification.

##### ADV\_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

##### ADV\_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

##### ADV\_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

##### ADV\_FSP.1.3c

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

##### ADV\_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

##### ADV\_FSP.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ADV\_FSP.1.2e**

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

---

**5.2.2 Guidance documents (AGD)****5.2.2.1 Operational user guidance (AGD\_OPE.1)**

---

**AGD\_OPE.1.1d**

The developer shall provide operational user guidance.

**AGD\_OPE.1.1c**

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

**AGD\_OPE.1.2c**

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

**AGD\_OPE.1.3c**

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

**AGD\_OPE.1.4c**

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

**AGD\_OPE.1.5c**

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

**AGD\_OPE.1.6c**

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

**AGD\_OPE.1.7c**

The operational user guidance shall be clear and reasonable.

**AGD\_OPE.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

---

**5.2.2.2 Preparative procedures (AGD\_PRE.1)**

---

**AGD\_PRE.1.1d**

The developer shall provide the TOE including its preparative procedures.

**AGD\_PRE.1.1c**

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

**AGD\_PRE.1.2c**

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**AGD\_PRE.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AGD\_PRE.1.2e**

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

**5.2.3 Life-cycle support (ALC)****5.2.3.1 Labelling of the TOE (ALC\_CMC.1)****ALC\_CMC.1.1d**

The developer shall provide the TOE and a reference for the TOE.

**ALC\_CMC.1.1c**

The TOE shall be labelled with its unique reference.

**ALC\_CMC.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**5.2.3.2 TOE CM coverage (ALC\_CMS.1)****ALC\_CMS.1.1d**

The developer shall provide a configuration list for the TOE.

**ALC\_CMS.1.1c**

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

**ALC\_CMS.1.2c**

The configuration list shall uniquely identify the configuration items.

**ALC\_CMS.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**5.2.4 Security Target (ASE)****5.2.4.1 Cryptographic operation (Hash Algorithm) (ASE\_TSS.1(c))****ASE\_TSS.1(c).1**

Refinement: The TOE summary specification shall describe how the TOE meets each SFR, including a proprietary Key Management Description (Appendix E), and [**Entropy Essay, 3rd party hardware components (including model/version numbers)**].

ASE\_TSS.1(c).1: Section 7 provides the TOE's Key Management Description, the separate Entropy Documentation and Analysis document provides the TOE's Entropy Essay, and the TOE includes the following 3<sup>rd</sup> party hardware components.

Component	Version/Part Number
Microcontroller	NXP ARM Cortex M4/M0 Processor P/N LPC4333
Account and Key Storage Chip	Infineon FM24V05-G
Entropy Chip	Microchip ATECC608B
HW Encryption Chip	Xilinx Ultrascale XCKU085 FPGA using AES XTS

**Table 3 3<sup>rd</sup> Party Hardware Components**

---

## 5.2.5 Tests (ATE)

### 5.2.5.1 Independent testing - conformance (ATE\_IND.1)

**ATE\_IND.1.1d**

The developer shall provide the TOE for testing.

**ATE\_IND.1.1c**

The TOE shall be suitable for testing.

**ATE\_IND.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ATE\_IND.1.2e**

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

---

## 5.2.6 Vulnerability assessment (AVA)

### 5.2.6.1 Vulnerability survey (AVA\_VAN.1)

**AVA\_VAN.1.1d**

The developer shall provide the TOE for testing.

**AVA\_VAN.1.1c**

The TOE shall be suitable for testing.

**AVA\_VAN.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AVA\_VAN.1.2e**

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

**AVA\_VAN.1.3e**

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

## 6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Security management
- Protection of the TSF

### 6.1 Cryptographic support

The Cryptographic support function satisfies the following security functional requirements:

- FDEAAcPP20E:FCS\_AFA\_EXT.1: The TOE supports a password authorization factor, and the password may be up to 64 characters (bytes) in length and can be composed of uppercase and lowercase letters and numbers.
- FDEAAcPP20E:FCS\_AFA\_EXT.2: The TOE does not have any power-saving states beyond power-off. After transitioning from the power-off to the power-on state, the user must authenticate before the TOE will allow data to be read from or written to the drive.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM.1(b): The TOE generates KEKs using its SHA-256 HMAC\_DRBG. The TOE does not store KEKs.
- FDEAAcPP20E:FCS\_CKM.1(c): The TOE can accept injection of externally supplied 256-bit DEKs. The TOE uses its cmkey interface to load the externally generated DEKs. The TOE AES-KW unwraps injected DEKs using the PSK.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM.4(a): When the TOE powers off (as the TOE has no other power states other than on and off), all values in normal SRAM memory drain to a zero state (as opposed to the FRAM—see below).
- FDEEEcPP20E:FCS\_CKM.4(b), FDEAAcPP20E:FCS\_CKM.4(d): The TOE includes a working memory, a separate Ferroelectric RAM (FRAM), and an entropy chip with internal EEPROM. The TOE includes working memory RAM as part of its microcontroller, and this serves as the working memory in which the TOE stores the AES-KW key (derived from the password and salt) and working copies of loaded DEKs (unwrapped during authentication). The TOE destroys these values when power is removed and when they are done being used.

The TOE possesses an FRAM in which the TOE stores the encrypted DEKs (the TOE can store 2 encrypted DEKs for the administrator). The TOE utilizes its entropy chip to store the PSK. Individual DEKs may be overwritten by the administrator loading a new DEK and specifying the key ID containing of key to be overwritten. Additionally, any operator may request that the TOE zeroize its keys, in which case, the TOE will overwrite its FRAM (one command) and entropy chip storage with (second command in this order) random data from the RBG, all zeros, and then all ones (in accordance with the DoD requirements for a standard erase) (note that the ones are not required but an extra overwrite to meet DoD requirements).

The TOE's internal microcontroller SRAM and FRAM are byte-addressable for both read and write operations. The TOE's entropy chip's internal EEPROM allows word addressing (4-byte access) as well as 32 bytes accessing, with a requirement that the 32-byte accesses being block aligned. However, the underlying EEPROM technology allows directly reading/writing of cells as 4-byte words (as opposed to a Flash technology in which reads/writes may occur on a 4-byte word access, while erase operations must occur only on some larger, sector/block basis).

- FDEEEcPP20E:FCS\_CKM.4(e): The TOE can, in effect, cryptographically erase all stored DEKs (in addition to directly clearing the encrypted DEKs) by destroying the salt value associated with the operator's

password. Once destroyed, no one can derive the AES-KW key needed to decrypt encrypted DEKs without the 256-bit salt value.

- FDEAAcPP20E/FDEAAcPP20E:FCS\_CKM\_EXT.4(a): The TOE considers keys that the operator has explicitly requested to be deleted as no longer necessary, and if zeroized, the TOE deems all keys other than the PSK as no longer needed. For those keys, the TOE will erase its hardware keystore (FRAM circuit).
- FDEAAcPP20E/FDEAAcPP20E:FCS\_CKM\_EXT.4(b): The TOE has the Compliant power saving state of G3 (Mechanical Off).
- FDEEEcPP20:FCS\_CKM\_EXT.6: The TOE clears its keys in accordance with FCS\_CKM.4(d).
- FCS\_COP.1: The TOE's Curtiss-Wright Crypto Firmware Library (version 2.00) contains cryptographic algorithm implementations that following NIST standards and has received the following CAVP algorithm certificates.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(a) (Verify)	ECDSA P-384 w/SHA-384 Verify	FIPS 186-4, ECDSA	<a href="#">A5103</a>
FCS_COP.1(b) (Hash)	SHA-256/384 Hashing	FIPS 180-4	<a href="#">A5103</a>
FCS_COP.1(c) (Keyed Hash)	HMAC-SHA-256/384	FIPS 198-1 & 180-4	<a href="#">A5103</a>
FCS_COP.1(d) (Key Wrap)	AES-256 KW	FIPS 197, SP 800-38F	<a href="#">A5103</a>
FCS_COP.1(f) (AES) FCS_CKM.1(b)	AES-256 XTS Encrypt/Decrypt	FIPS 197	<a href="#">A5793</a>
FCS_RBG_EXT.1 (Random)	SHA-256 HMAC_DRBG	SP 800-90A	<a href="#">A5103</a>

**Table 4 Cryptographic Algorithms**

- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(a): The TOE utilizes ECDSA P-384 with SHA-384 signatures to verify the authenticity of firmware updates. Upon receiving a candidate update and the accompanying signature file, the TOE uses an embedded public key to verify the ECDSA signature against the received image. The verification uses SHA-384 and follows the FIPS 186-4 ECDSA format.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(b): The TOE implements the SHA-256 and SHA-384 algorithms and uses the SHA-256 algorithm as part of PBKDFv2 key derivation and also as part of its HMAC\_DRBG. The TOE uses SHA-384 hashing when verifying trusted update ECDSA P-384 signatures and then calculating HMAC-SHA-384 checksums.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP(c): The TOE implements HMAC-SHA-256/384 using a 256/384-bit key, the SHA-256/384 hash algorithm, a 512/1024-bit block size, and an output MAC length of 256/384-bits. The TOE uses its HMAC-SHA-256 implementation during PBKDFv2 key derivation and during HMAC\_DRBG random number generation. The TOE uses its HMAC-SHA-384 implementation to ensure the integrity of the imported DEK, as well as to augment admin authentication.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(d): The TOE uses AES-KW (compliant with NIST SP 800-38F) to encrypt the DEKs stored in FRAM and to unwrap DEKs injected by an administrator.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(f): The TOE possesses an AES XTS implementation dedicated to drive encryption/decryption. The TOE's implementation exclusively uses 256-bit keys.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_KDF\_EXT.1: The TOE uses 800-132 PBKDF in counter mode using SHA-256 and 1000 iterations and a 256-bit salt to transform the operator's password into a key for wrapping/unwrapping.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_KYC\_EXT.1/2: The TOE uses PBKDFv2 to transform the operator's password into a 256-bit BEV, and then uses that BEV to AES-KW unwrap the DEKs stored in the TOE hardware key store. The AES-KW unwrap operation will verify whether the operator supplied the correct password. If the operator supplied the correct password, then the TOE will have access to the DEK values (in memory).
- FDEAAcPP20E:FCS\_PCC\_EXT.1: The TOE allows passwords up to 64-bytes (characters in length), and the TOE checks to only allow uppercase/lowercase letters and numbers. The TOE will reject a password



containing other characters. The TOE conditions passwords by combining them with a 256-bit salt using PBKDFv2.

- FDEAAcPP20E/FDEEEcPP20E:FCS\_RBG\_EXT.1: The TOE includes an SHA-256 HMAC\_DRBG that it seeds with at least 256-bits of entropy from a hardware-based noise source.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_SNI\_EXT.1: The TOE generates its salts (the admin account has a 256-bit salt used during PBKDFv2 derivation) using its SHA-256 HMAC\_DRBG. The TOE generates nonces but generates its 256-bit AES XTS tweaks using encrypted logical address block values.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_VAL\_EXT.1: A password is required when the machine is power cycled. The TOE validates the operator's password by attempting an AES-KW decrypt/unwrap operation. If the AES-KW decryption operation fails, then the TOE treats this as an invalid login and increments its failed login attempts counter.

If the counter reaches five, the TOE enters an error state and fails to accept any new passwords. It needs to be rebooted.

## 6.2 User data protection

The User data protection function satisfies the following security functional requirements:

- FDEEEcPP20E:FDP\_DSK\_EXT.1: The TOE provides hardware-based FDE and encrypts the entirety of the drive through an AES-256 XTS block based encryption. The TOE sits as an In-Line Encryptor (ILE) in the NVMe path between the host and the drives. Because of its position, the ILE guarantees that all data written to and read from the drive is encrypted. The User Guide describes the TOE's initialization process and setup for the HW-layer. The HW-layer performs block-based encryption of the entire drive leaving no sectors/blocks unencrypted.

## 6.3 Security management

The Security management function satisfies the following security functional requirements:

- FDEAAcPP20E:FMT\_MOF.1: The TOE claims no Compliant power saving states beyond power off. The TOE does not allow administrators or users to manage or configure the Compliant power saving states supported by the TOE.
- FDEAAcPP20E/FDEEEcPP20E:FMT\_SMF.1: The TOE allows an administrator to change a DEK, overwrite a DEK, cryptographically erase all DEKs, import a wrapped DEK using the cmkey command. The TOE supports changing of the authorization factor (the administrator can zeroize the TOE and then reinitialize the TOE to change the associated password) using the cmkey command to zeroize and the cminit command to set a new password.
- FDEAAcPP20E:FMT\_SMR.1 – The TOE maintains an administer role that can administer the TOE. The only users are administrators.

## 6.4 Protection of the TSF

The Protection of the TSF function satisfies the following security functional requirements:

- FDEEEcPP20E:FPT\_FUA\_EXT.1: The TOE uses an internal ECDSA P-384 public key (hardcoded within the TOE's existing microprocessor firmware image, stored within the microprocessor) to verify new firmware images before writing the firmware to the TOE's internal storage.
- FDEAAcPP20E/FDEEEcPP20E:FPT\_KYP\_EXT.1: The TOE stores keys in its FRAM chip (part of its hardware keystore), which acts like non-volatile memory. The TOE stores all keys in encrypted form (encrypted with an AES-KW key derived from the operator's password plus an internal salt), with the exception of the Pre-Shared Key (PSK). The PSK exists to protect the import of the DEK into the TOE, and

protect the export of the user token which is used to implement an optional second factor for user login. The user uses their protected copy of the PSK to wrap a DEK for import into the TOE, and the TOE decrypts the wrapped DEK internally before loading into the crypto engine, or rewrapping using a conditioned password and then storing in non-volatile storage. The PSK does not participate in the key chain protecting DEKs, but instead, only provides an extra layer of transient protection during DEK injection.

- FDEAAcPP20E/FDEEEcPP20E:FPT\_PWR\_EXT.1/2: The TOE provides the Compliant power-saving state G3, mechanical off. The TOE does not allow administrators or users to manage or configure the Compliant power saving states supported by the TOE. The TOE must be fully rebooted from this state.
- FDEAAcPP20E/FDEEEcPP20E:FPT\_TST\_EXT.1: The TOE includes the following power-up Known Answer Tests (KATs) to ensure that each of its cryptographic algorithms operates correctly.
  - ECDSA verify test
  - SHA-256/384 hashing tests
  - HMAC-SHA-256/384 hashing test
  - AES-ECB encryption/decryption test
  - AES256 keywrap/keyunwrap test
  - AES-XTS encryption/decryption test
  - SHA-256 HMAC\_DRBG tests (including SP 800-90A section 11.3 health tests)
  - PBKDF2 hash test
  - integrity test
- FDEAAcPP20E/FDEEEcPP20E:FPT\_TUD\_EXT.1: The TOE allows field updates to the HW-layer's firmware that have been signed and delivered by Curtiss Wright. The TOE will verify the ECDSA P-384 with SHA-384 signature of the update image, and if valid, the TOE will update its firmware. The HTTPS interface must be used for updates.

## 7. Key Management Description

The key management description explains each key, cryptomodule and overall encryption architecture. Each key is identified in the table below.

Key Identifier	Storage Location	How Key Protected	How key Derived/Generated	Strength of Key	When Key Destroyed
User Passphrase	Memory - transient	N/A	N/A	N/A	Immediately after use
Wrap Key	Memory	N/A	The TOE uses 800-132 PBKDF using HMAC-SHA-256 and a number of iterations and a 256-bit salt to transform the operator's password into a Wrap Key	256 bits	Upon power-cycle
Data Encryption Key (DEK)	FRAM	AES KW Encrypted	Injected from outside (wrapped with PSK)	256 bits	Upon overwrite or upon zeroization
Pre-Shared Key (PSK)	Entropy Chip	Plaintext	Injected during manufacturing	256 bits	Upon destructive zeroization
Key Encryption Key (KEK)	Memory	N/A	Generated from approved DRBG	256 bits	Immediately after use
User Authentication Token	FRAM	AES KW Encrypted	Generated from approved DRBG	256 bits	Upon zeroization

**Table 5 Key Identification**

The TOE's encryption engine was custom developed by the vendor to provide HW-based full drive encryption. The TOE consists of a microcontroller with custom firmware and HW components. The microcontroller handles the operator authentication, key management, and firmware update verification.

The HW component receives the DEK from the microcontroller and then encrypts/decrypts data written to/read from the encrypted partition/drive. The TOE encrypts the keys its internal, dedicated FRAM memory, it provides no access to this memory, and only exposes the encrypted Flash Storage Module (drive) to network-attached clients. The TOE ensures that access to the FSM/drives is always encrypted, and does not permit plaintext access to protected partitions or drives. Because the TOE utilizes a dedicated processor and dedicated internal FRAM, the TOE only provides access to the FSM/drives once fully initialized and after receiving the administrator's password.

The TOE uses the following cryptographic implementations:

1. HW AES-XTS encryption/decryption – a programmable hardware chip to accelerate full drive encryption operations.
2. Curtiss-Wright microcontroller cryptographic implementation - used for all other cryptographic needs (authentication, key management, and trusted update image verifications)