

---

# **Ampex Software Full Drive Encryption Security Target**

Version 0.5  
08/11/25

---

*Prepared for:*

**Ampex Data Systems Corporation**

26460 Corporate Avenue  
Hayward, CA 94545

*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) .....	15
5.1.3 Security management (FMT) .....	15
5.1.4 Protection of the TSF (FPT) .....	16
5.2 TOE SECURITY ASSURANCE REQUIREMENTS .....	17
5.2.1 Development (ADV) .....	18
5.2.2 Guidance documents (AGD) .....	18
5.2.3 Life-cycle support (ALC) .....	19
5.2.4 Tests (ATE) .....	19
5.2.5 Vulnerability assessment (AVA) .....	20
5.2.6 Security Target (ASE) .....	20
<b>6. TOE SUMMARY SPECIFICATION .....</b>	<b>21</b>
6.1 CRYPTOGRAPHIC SUPPORT .....	21
6.2 USER DATA PROTECTION .....	23
6.3 SECURITY MANAGEMENT .....	23
6.4 PROTECTION OF THE TSF .....	23
<b>7. KEY MANAGEMENT DESCRIPTION .....</b>	<b>25</b>

## LIST OF TABLES

Table 1 Technical Decisions .....	6
Table 2 TOE Security Functional Components .....	10
Table 3 Assurance Components .....	17
Table 4 SW-FDE OpenSSL Cryptographic Algorithms .....	22
Table 5 SW-FDE Kernel Cryptographic Algorithms .....	22
Table 6 Key Identification .....	25

## 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 Ampex Software Full Drive Encryption provided by Ampex Data Systems Corporation. The TOE is being evaluated as a full drive encryption.

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.
  - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration may be 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. Alternately, a usually descriptive textual extension may be added after a slash (/) character to identify a specific iteration. For example, iterations of a requirement such as FCS\_COP.1 might be identified as FCS\_COP.1/HASH and FCS\_COP.1/CRYPT.
  - 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 "... ~~some~~ **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** – Ampex Software Full Drive Encryption Security Target

**ST Version** – Version 0.5

**ST Date** – 08/11/25

### 1.2 TOE Reference

**TOE Identification** – Ampex Software Full Drive Encryption

**TOE Developer** – Ampex Data Systems Corporation

**Evaluation Sponsor** – Ampex Data Systems Corporation

### 1.3 TOE Overview

The Target of Evaluation (TOE) is Ampex Software Full Drive Encryption (Ampex SW FDE) version 1.10.

The TOE provides software Full Drive Encryption of removable drives within the computing system in which it operates.

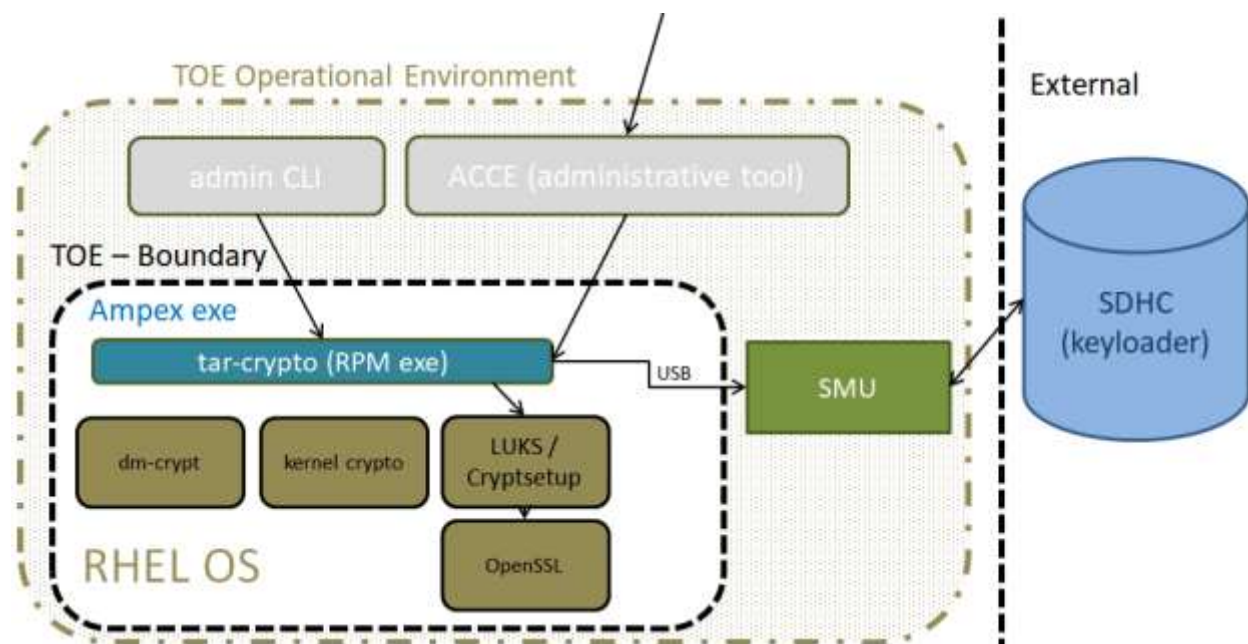
### 1.4 TOE Description

The Ampex SW FDE is a software package that installs into a computing system running Red Hat Enterprise Linux (RHEL). The Product protects independent data drives (as opposed to the system (OS) drive) and accepts a user passphrase or an authorization factor from an external USB Drive, validates the authorization/authentication factor, and if correct utilizes it to unlock the Data Encryption Key used to encrypt/decrypt derive the removable drive.

The TOE has been tested on two different computing platforms both running Red Hat Enterprise Linux 8.8 executing on an Intel Xeon D-1746TER (Ice Lake/Sunny Cove) CPU. One of the computing platforms included the hardware component (the SMU) which allows use of a USB drive authentication factor, while the other platform lacks such hardware and accepts only a password authentication factor to unlock the FDE.

#### 1.4.1 TOE Architecture

The Product contains software that, once installed, leverages RHEL OS components to provide FDE protection for persistent data stored to a removable drive.



##### 1.4.1.1 Physical Boundaries

The TOE is software, but inherits the physical perimeter of the computing platform upon which it executes.

##### 1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Ampex SW FDE:

- 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 all partitions on the removable drive (so that no plaintext exists) and does so without user intervention.

---

#### 1.4.1.2.3 Security management

---

The TOE provides each of required management services to manage the full drive encryption using a locally accessed Command Line Interface (CLI).

---

#### 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 fail, the TOE will not go into an operational mode.

---

### 1.4.2 TOE Documentation

---

Ampex Data Systems Corporation SW Encryption Layer Certifiable Encryption, Document No. 8470200-010, Revision 1.2.0, Date Created July 15, 2025 [**Admin 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, Version 3.1, Revision 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
  - Part 3 Conformant
- Package Claims:
  - collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, Version 2.0 + Errata 20190201, 01 February 2019 (FDEAAcPP20E)
  - collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201, 01 February 2019 (FDEEEcPP20E)

Package	Technical Decision	Applied	Notes
CPP_FDE_AA_V2.0E	TD0929 - FIT Technical Decision: Clarification to FCS_PCC_EXT.1.1	Yes	
CPP_FDE_AA_V2.0E	TD0769 - FIT Technical Decision for FPT_KYP_EXT.1.1	Yes	
CPP_FDE_AA_V2.0E	TD0767 - FIT Technical Decision for FMT_SMF.1.1	Yes	
CPP_FDE_AA_V2.0E	TD0766 - FIT Technical Decision for FCS_CKM.4(d) Test Notes	Yes	
CPP_FDE_AA_V2.0E	TD0765 - FIT Technical Decision for FMT_MOF.1	Yes	
CPP_FDE_AA_V2.0E	TD0760 - FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	Yes	
CPP_FDE_AA_V2.0E	TD0759 - FIT Technical Decision for FCS_AFA_EXT.1.1	Yes	
CPP_FDE_AA_V2.0E	TD0606 - FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No	Product not a NAS
CPP_FDE_AA_V2.0E	TD0458 - FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	
CPP_FDE_EE_V2.0E	TD0769 - FIT Technical Decision for FPT_KYP_EXT.1.1	Yes	
CPP_FDE_EE_V2.0E	TD0766 - FIT Technical Decision for FCS_CKM.4(d) Test Notes	No	SFR not selected
CPP_FDE_EE_V2.0E	TD0606 - FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No	Product not a NAS
CPP_FDE_EE_V2.0E	TD0464 - FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Yes	
CPP_FDE_EE_V2.0E	TD0460 - FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Yes	
CPP_FDE_EE_V2.0E	TD0458 - FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	

**Table 1 Technical Decisions**

### 2.1 Conformance Rationale

The ST conforms to the FDEAAcPP20E/FDEEEcPP20E. The security problem definition, security objectives, and security requirements are referenced and available in the PPs listed above.

### 3. Security Objectives

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

In general, the FDEAAcPP20E/FDEEEcPP20E has defined Security Objectives appropriate for full drive encryption and as such are applicable to the Amplex Software Full Drive Encryption 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 FDEAAcPP20E/FDEEEcPP20E. The FDEAAcPP20E/FDEEEcPP20E defines the following extended requirements and since they are not redefined in this ST the FDEAAcPP20E/FDEEEcPP20E should be consulted for more information in regard to those CC extensions.

### Extended SFRs:

- FDEAAcPP20E:FCS\_AFA\_EXT.1: Authorization Factor Acquisition - per TD0759
- 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
- 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 - per TD0929
- FDEAAcPP20E:FCS\_RBG\_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FDEAAcPP20E:FCS\_SNI\_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) - per TD0760
- FDEEEcPP20E:FCS\_SNI\_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- 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 - per TD0769
- FDEEEcPP20E:FPT\_KYP\_EXT.1: Protection of Key and Key Material - per TD0769
- 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 FDEAAcPP20E/FDEEEcPP20E. The refinements and operations already performed in the FDEAAcPP20E/FDEEEcPP20E are not identified (e.g., highlighted) here, rather the requirements have been copied from the FDEAAcPP20E/FDEEEcPP20E and any residual operations have been completed herein. Of particular note, the FDEAAcPP20E/FDEEEcPP20E 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 FDEAAcPP20E/FDEEEcPP20E. The FDEAAcPP20E/FDEEEcPP20E should be consulted for the assurance activity definitions.

### 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Ampex Software Full Drive Encryption TOE.

Requirement Class	Requirement Component
<b>FCS: Cryptographic support</b>	FDEAAcPP20E:FCS_AFA_EXT.1: Authorization Factor Acquisition - per TD0759
	FDEAAcPP20E:FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition
	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)
	FDEAAcPP20E:FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage) - per TD0766
	FDEEEcPP20E:FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage) - per TD0766
	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(f): Cryptographic operation (AES Data Encryption/Decryption) - per TD0760
	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)

Requirement Class	Requirement Component
	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 - per TD0929
	FDEAAcPP20E:FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FDEAAcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) - per TD0760
	FDEEEcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
	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 - per TD0765
	FDEAAcPP20E:FMT_SMF.1: Specification of Management Functions - per TD0767
	FDEEEcPP20E:FMT_SMF.1: Specification of Management Functions
	FDEAAcPP20E:FMT_SMR.1: Security Roles
<b>FPT: Protection of the TSF</b>	FDEAAcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material - per TD0769
	FDEEEcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material - per TD0769
	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 2 TOE Security Functional Components**

### 5.1.1 Cryptographic support (FCS)

#### 5.1.1.1 Authorization Factor Acquisition - per TD0759 (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,*  
*- an external USB token factor that is at least the same security strength as the BEV and is providing a submask generated by the TOE using the RBG as specified in FCS\_RBG\_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) (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.

---

**5.1.1.4 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 [- *generate a DEK using the RBG as specified in FCS\_RBG\_EXT.1*] and specified cryptographic key sizes [256 bits].

---

**5.1.1.5 Cryptographic Key Destruction (Power Management) (FDEAAcPP20E:FCS\_CKM.4(a))**

---

**FDEAAcPP20E:FCS\_CKM.4.1(a)**

Refinement: The TSF shall [*instruct the Operational Environment to clear*] 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).

---

**5.1.1.6 Cryptographic Key Destruction (Power Management) (FDEEEcPP20E:FCS\_CKM.4(a))**

---

**FDEEEcPP20E:FCS\_CKM.4.1(a)**

The TSF shall [*instruct the Operational Environment to clear*] 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.

---

**5.1.1.7 Cryptographic Key Destruction (Software TOE, 3rd Party Storage) - per TD0766 (FDEAAcPP20E:FCS\_CKM.4(d))**

---

**FDEAAcPP20E: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 [removal of power to the memory],*  
- *For non-volatile storage that consists of the invocation of an interface provided by the underlying platform that [o logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes]]*  
that meets the following: no standard.

---

**5.1.1.8 Cryptographic Key Destruction (Software TOE, 3rd Party Storage) - per TD0766 (FDEEEcPP20E:FCS\_CKM.4(d))**

---

**FDEEEcPP20E: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 [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 [zeroes]]*  
that meets the following: no standard.

#### 5.1.1.9 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.

#### 5.1.1.10 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.

#### 5.1.1.11 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.

#### 5.1.1.12 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.

#### 5.1.1.13 Cryptographic Key Destruction Types (FDEEEcPP20E:FCS\_CKM\_EXT.6)

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

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

#### 5.1.1.14 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:

[*Section 6 and Appendix D, Implementing 'NIST curves' [P-384]; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes*].

#### 5.1.1.15 Cryptographic Operation (Signature Verification) (FDEEEcPP20E:FCS\_COP.1(a))

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

The TSF shall perform cryptographic signature services (verification) in accordance with a [*Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater*] that meets the following: [*FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-384]; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes*].

#### 5.1.1.16 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-384*] that meet the following: ISO/IEC 10118-3:2004.

**5.1.1.17 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-384*] that meet the following: ISO/IEC 10118-3:2004.

**5.1.1.18 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-384*] and cryptographic key sizes [*384*] that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

**5.1.1.19 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-384*] and cryptographic key sizes [*384*] that meet the following: [*NIST SP 800-38B*].

**5.1.1.20 Cryptographic operation (AES Data Encryption/Decryption) - per TD0760 (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*].

**5.1.1.21 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*].

**5.1.1.22 Cryptographic operation (Key Encryption) (FDEAAcPP20E:FCS\_COP.1(g))****FDEAAcPP20E:FCS\_COP.1.1(g)**

Refinement: The TSF shall perform key encryption and decryption in accordance with a specified cryptographic algorithm AES used in [*CBC*] mode and cryptographic key sizes [*256 bits*] that meet the following: AES as specified in ISO /IEC 18033-3, [*CBC as specified in ISO/IEC 10116*].

**5.1.1.23 Cryptographic Operation (Key Encryption) (FDEEEcPP20E:FCS\_COP.1(g))****FDEEEcPP20E:FCS\_COP.1.1(g)**

Refinement: The TSF shall perform key encryption and decryption in accordance with a specified cryptographic algorithm AES used in [*CBC*] mode and cryptographic key sizes [*256 bits*] that meet the following: AES as specified in ISO/IEC 18033-3, [*CBC as specified in ISO/IEC 10116*].

**5.1.1.24 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.

#### 5.1.1.25 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 [256 bits] for asymmetric keys.

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

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

#### 5.1.1.26 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 derivation as specified in FCS\_KDF\_EXT.1, - key encryption as specified in FCS\_COP.1(g)* ] while maintaining an effective strength of [256 bits] for symmetric keys and an effective strength of [*not applicable*] for asymmetric keys.

#### 5.1.1.27 Cryptographic Password Construct and Conditioning - per TD0929 (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 at least [64] characters in the set of upper case characters, lower case characters, numbers, and [all ASCII printable characters] and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[SHA-384], with [2774010] iterations], and output cryptographic key sizes [256 bits] that meet the following: NIST SP 800-132. (TD0929 applied)

#### 5.1.1.28 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 [ISO/IEC 18031:2011] using [CTR\_DRBG (AES)].

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

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*none*] *software-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.

#### 5.1.1.29 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) - per TD0760 (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 [*unique nonces with a minimum size of 64 bits*].

##### FDEAAcPP20E:FCS\_SNI\_EXT.1.3

The TSF shall [*create IVs in the following matter* ]  
 - *CBC: IVs shall be non-repeating and unpredictable,*  
 - *XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer*]. (TD0760 applied)



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

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

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

#### FDEEEcPP20E:FCS\_SNI\_EXT.1.3

The TSF shall create IVs in the following manner [

- *CBC: IVs shall be non-repeating and unpredictable,*
- *XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer*].

### 5.1.1.31 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): [*hash the BEV as specified in [FCS\_COP.1(c)] and compare it to a stored hashed value*]

#### 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 [*institute a delay such that only [43,200] can be made within a 24 hour period*]

## 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 - per TD0765 (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 - per TD0767 (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) [*no other functions*]  
(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) [*no other functions*].

### 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 Protection of Key and Key Material - per TD0769 (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)*].  
(TD0769 applied)

#### 5.1.4.2 Protection of Key and Key Material - per TD0769 (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)*].  
(TD0769 applied)

#### 5.1.4.3 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**].

#### 5.1.4.4 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)

#### 5.1.4.5 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,  
[*system shutdown*].

#### 5.1.4.6 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, [*system shutdown*].



#### 5.1.4.7 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 (including ECDSA verify test, SHA2 hashing tests, HMAC-SHA tests, AES-256 CBC/XTS encrypt/decrypt test, and AES-256 CTR\_DRBG test) and health test].

#### 5.1.4.8 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 (including ECDSA verify test, SHA2 hashing tests, HMAC-SHA tests, AES-256 CBC/XTS encrypt/decrypt test, and AES-256 CTR\_DRBG test) and health test].

#### 5.1.4.9 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 [*software*].

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

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

##### 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.

#### 5.1.4.10 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 [*software*].

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

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

##### FDEEEcPP20E: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.

## 5.2 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
ADV: Development	ADV FSP.1: Basic Functional Specification
AGD: Guidance documents	AGD OPE.1: Operational User Guidance
	AGD PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC CMC.1: Labelling of the TOE
	ALC CMS.1: TOE CM Coverage
ATE: Tests	ATE IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA VAN.1: Vulnerability Survey
ASE: Security Target	ASE TSS.1: Security Target

Table 3 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 categorization 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 fulfill 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 Tests (ATE)**

### **5.2.4.1 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.

---

**5.2.5 Vulnerability assessment (AVA)**

---

**5.2.5.1 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.

---

**5.2.6 Security Target (ASE)**

---

**5.2.6.1 Security Target (ASE\_TSS.1)**

---

**ASE\_TSS.1.1c**

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

## 6. TOE Summary Specification

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 password authorization factor, and the password may be between 15 and (up to) 64 characters in length and can be composed of all ASCII printable characters. The TOE also supports an External USB factor generated by the TOE (separately executing on a ground/provisioning station).
- FDEAAcPP20E:FCS\_AFA\_EXT.2: The TOE does not have any power-saving states beyond power-on and power-off. After transitioning from the power-off to the power-on state, the user must provide authentication factors before the TOE will allow data to be read from or written to the drive.
- FDEEEcPP20E:FCS\_CKM.1(b)/(c): The TOE generates 256-bit DEKs using its AES-256 CTR\_DRBG. Because the DRBG has a security strength of 256 bits, the DEKs generated are sufficient for the TOE's 256-bit AES data encryption/decryption. The TOE stores these keys encrypted in dedicated headers on the drives (in the first few megabytes of an un-partitioned, drives or at the start of each partition).
- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM.4(a): When the TOE powers off (as the TOE has no other power states other than on and off (G3)), all values in memory drain to a zero state.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM.4(d): The computing platform upon which the TOE executes has RAM, and this serves as the working memory in which the TOE temporarily stores working copies of key material (for example, the Derived Key [DerKey], which is derived from the user's password and salt using PBKDFv2 and the DEKs currently in use (if any). The TOE clears keys from memory by a removal of power.

Additionally, the TOE stores encrypted DEKs in a header for the encrypted drive partitions. The TOE clears these keys by through an internal call which overwrites the data with zeroes.

- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM\_EXT.4(a): The TOE clears the DerKey and DEKs from userspace memory immediately after the operation for which it is needed, while DEKs will be held in kernel memory while the drive is accessible. If the user reboots or powers off the system, then the TOE will clear any in-use DEKs from kernel memory.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_CKM\_EXT.4(b): The TOE has no Compliant power saving states other than power on and off (G3).
- FDEEEcPP20E:FCS\_CKM\_EXT.6: The TOE clears its keys in accordance with FCS\_CKM.4(d).
- FCS\_COP.1: The TOE performs cryptographic algorithms in accordance with the following NIST standards and has received the following CAVP algorithm certificates.

The TOE uses its cryptographic library (OpenSSL [rhel8.20231130](#) [based on 1.1.1] for RHEL 8.8) to support all of its FDE functionality.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(a) (Verify)	ECDSA P-384 w/ SHA-384 Verify	FIPS 186-4, ECDSA	<a href="#">A7170</a>
FCS_COP.1(b) (Hash)	SHA-384 Hashing	FIPS 180-4	<a href="#">A7170</a>
FCS_COP.1(c) (Keyed Hash)	HMAC-SHA-384	FIPS 198-1 & 180-4	<a href="#">A7170</a>

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(g) (AES)	AES-256 CBC Encrypt/Decrypt	FIPS 197	<a href="#">A7170</a>
FCS_RBG_EXT.1 (Random)	AES-256 CTR_DRBG	SP 800-90A	<a href="#">A7170</a>
FCS_CKM.1(b) (Key Gen)			
FCS_CKM.1(c) (Key Gen)			

**Table 4 SW-FDE OpenSSL Cryptographic Algorithms**

The TOE also uses a 4.18 kernel Cryptography for the following bulk encryption/decryption algorithms.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(f) (AES)	AES-256 XTS Encrypt/Decrypt	FIPS 197	<a href="#">A7171</a>

**Table 5 SW-FDE Kernel Cryptographic Algorithms**

- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(a): The TOE utilizes ECDSA P-384 w/ SHA-384 to verify the authenticity of firmware updates. Upon receiving a candidate update and the accompanying signature file, the TOE uses an embedded public key (see FPT\_TUD\_EXT.1 below for the location) to verify the signature against the received image. The verification uses SHA-384 following the FIPS 186-4 ECDSA format.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(b): The TOE's OpenSSL library provides the SHA-384 algorithm and uses the algorithm as part of PBKDFv2 password-based key derivation, and trusted update signature verification respectively.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(c): The TOE implements HMAC-SHA-384 using 256-bit keys, the SHA-384 hash algorithm, a 1024-bit block size, and an output MAC length of 384 bits. The TOE uses HMAC-SHA-384 as part of all PBKDF (NIST SP 800-132) key management operations.
- FDEEEcPP20E:FCS\_COP.1(f): The TOE uses an AES XTS cryptographic implementation dedicated to drive encryption/decryption. This implementation uses AES- 256 bit keys.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_COP.1(g): The TOE has an AES CBC implementation used for key managements operations (decryption of the encrypted DEKs). This implementation uses AES- 256 bit keys.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_KDF\_EXT.1: The TOE uses 800-132 (PBKDFv2) with HMAC-SHA-384 and a number of iterations and a 256 bit salt to transform the operator's password into a Derived Key for decrypting the encrypted DEKs. The number of iterations is determined by the specified number of milliseconds (2000 milliseconds) multiplied by the number of PBKDF operations per/second to achieve a delay specified by the administrator. The TOE has an empirically determined benchmark for PBKDF2 of 1,387,005 iterations per second for a 256-bit key, and thus the TOE uses a count of 2,774,010 iterations to achieve a 2 second delay.
- 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 decrypt the DEKs stored in the header(s) stored on the drive.
- FDEAAcPP20E:FCS\_PCC\_EXT.1: The TOE allows passwords up to 64 characters in length, and the TOE allows uppercase/lowercase letters, numbers, and ASCII printable characters. The TOE will reject any password longer than 64 characters. The TOE will reject a password containing other characters. The TOE conditions passwords by combining them with a 256-bit salt using PBKDFv2.
- FDEAAcPP20E:FCS\_RBG\_EXT.1: The TOE includes an AES-256 CTR\_DRBG that it seeds with at least 256-bits of entropy from a software-based noise source.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_SNI\_EXT.1: The TOE generates its salts and AES-CBC IVs using its AES-256 CTR\_DRBG. The TOE generates no nonces but generates it 256-bit AES XTS tweaks (used for data partition encryption) also using its CTR\_DRBG.
- FDEAAcPP20E/FDEEEcPP20E:FCS\_VAL\_EXT.1: The TOE validates the operator's password by first subjecting the password and salt to PBKDFv2 to form the Derived Key (DerKey). The TOE uses the DerKey to decrypt the masterKey stripes and reconstitutes the masterKey; however, before using the masterKey, the TOE first performs iterative HMAC-SHA-384 using the operator's password, the masterKey salt, masterKey

iterations, and masterKey as inputs, and then compares the resulting value to the stored masterKey's digest stored in the header to ensure the two match.

If the TOE institutes a delay of 2 seconds on each authentication attempt, such that one cannot attempt more than 43,200 authentication attempts per 24 hour period.

## 6.2 User data protection

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

- FDEEEcPP20E:FDP\_DSK\_EXT.1: The TOE provides FDE that encrypts the entirety of the drive or drive partitions through AES-XTS block based encryption. The Admin Guide describes the TOE's initialization process and setup for the SW-FDE. The vendor designed the TOE to execute from a separate, unencrypted, system drive where the RHEL-operating system resides. That system drive contains no user data, as the TOE stores all user data on the removable drive that the TOE encrypts. When the administrator configures a removable drive for use, the TOE encrypts the entire drive (with a small area reserved for the Linux Unified Key Setup (LUKS) header). There is no interface to access user data before the entire encryption process is complete.

## 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. Only the authorized administrator can issue the shutdown command.
- FDEAAcPP20E/FDEEEcPP20E:FMT\_SMF.1: The TOE provides each of the required management services with no additional ones. Because the TOE fulfills the AA and EE requirements together, the TOE need not "forward" requests to change the DEK or cryptographically erase the DEK. The TOE supports changing of the authorization factors (the administrator can remove a partition and recreate it to change the associated password). The Admin Guide describes updating the TOE's software through loading and installation of RPM updates (the TOE relies upon RHEL OS to verify the signature, and (if the signature verifies successfully) install the RPM update). The TOE does not provide any manageable power-saving states.
- FDEAAcPP20E:FMT\_SMR.1 – The TOE's RHEL OS maintains an administer role that can administer the TOE.

## 6.4 Protection of the TSF

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

- FDEAAcPP20E/FDEEEcPP20E:FPT\_KYP\_EXT.1: The TOE stores encrypted DEKs in the header of each drive partition.
- FDEAAcPP20E/FDEEEcPP20E:FPT\_PWR\_EXT.1/2: The TOE provides the compliant power-saving state power-off and G3, mechanical off. The TOE enters this state when the operator shuts off the device. 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
  - SHA2 hashing tests
  - HMAC-SHA tests
  - AES-256 CBC/XTS encrypt/decrypt test
  - AES-256 CTR\_DRBG test

The TOE also performs entropy health tests (at start up).

There are no non-cryptographic functions that affect the correct operation of the TSF.

- FDEAAcPP20E/FDEEEcPP20E:FPT\_TUD\_EXT.1: The TOE can display its current software version (queried using RHEL RPM commands) and an administrator can update the TOE software using signed RPM update files. The TOE's RHEL OS will verify the signature on the RPM software update files (using its cryptographic library in conjunction with the vendor's signature key, stored in the RPM database in /var/lib/rpm, to verify the ECDSA P-384 with SHA-384 signature) before installing it, and will reject any update with an invalid signature.



## 7. Key Management Description

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

Key Identifier	Storage Location	How Key Protected	How key Derived	Strength of Key	When Key Destroyed
User Passphrase	Memory - transient	N/A	N/A	N/A	Immediately after use
USB Factor	Memory - transient	N/A	N/A	N/A	Immediately after use
Derived Key	Memory - transient	N/A	The TOE uses 800-132 KDF in counter mode using HMAC-SHA-384 and a number of iterations and a 256 bit salt to transform the operator's password into a Derived Key	256 bits	Immediately after use
DEK (Master Key)	Memory - transient	N/A	Generated from approved DRBG	256 bits	Immediately after use
DEK (Master Key)	Partition Header - persistent	AES CBC Encrypted	Generated from approved DRBG	256 bits	When partition closed or when partition no longer encrypted

**Table 6 Key Identification**

Keys are destroyed through zeroization or overwrite when no longer needed. The data encryption engine is based on LUKS, and is comprised of both a userspace component and a kernel-level component. The userspace component handles derivation of the Derived Key from the user's password and the subsequent decryption of the Master Key (DEK) with the Derived Key. The kernel-level component receives the DEK from the userspace component and then encrypts/decrypts data written to/read from the encrypted partition/drive. The TOE uses PBKDFv2 to transform the operator's password into a 256-bit BEV, and then uses that BEV to AES decrypt a Master Key (DEK) stored in the header stored on the drive. The data encryption software runs atop the RHEL 8.8 software (running on a hardware computing platform), where the software executes on the main processor, with the software stored in a dedicated OS/system drive. While the TOE does not encrypt its system drive, the TOE encrypts a removable drive. TOE only provides access to the encrypted data once fully initialized and after receiving the administrator's password or USB authentication factor.

The SW-FDE cryptographic implementation uses cryptographic from three crypto modules:

1. OpenSSL – used for all LUKS key management (but not encrypting/decrypting drive data) and for verification of trusted updates.
2. Kernel – used for encrypting data on the partition.