Curtiss-Wright Defense Solutions Compact Network Storage 4-Slot Software Encryption Layer (FDEEEcPP20E/FDEAAcPP20E) Security Target

Version 0.5
04/15/2019

Prepared for:

Curtiss-Wright Defense Solutions

2600 Paramount Pl #200
Fairborn, OH 45324

Prepared By:

Gossamer Laboratories

www.gossamersec.com

LIST OF TABLES

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 Compact Network Storage 4-Slot Software Encryption Layer provided by Curtiss-Wright Defense Solutions. The TOE is being evaluated as a 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.
  - 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 "… ~~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 – Curtiss-Wright Defense Solutions Compact Network Storage 4-Slot Software Encryption Layer (FDEEEcPP20E/FDEAAcPP20E) Security Target

ST Version – Version 0.5

ST Date – 04/15/2019

1.2 TOE Reference

TOE Identification – Curtiss-Wright Defense Solutions Curtiss-Wright Defense Solutions Compact Network Storage 4-Slot Software Encryption Layer

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 Compact Network Storage 4-Slot Software Encryption Layer.

The TOE provides software Full Drive Encryption of removable drives.

1.4 TOE Description

The CNS4 Software Encryption Layer (hereafter referred to as the TOE) is a rugged Network Attached Storage (NAS) file server for use in Unmanned Aerial Vehicles (UAV), Unmanned Underwater Vehicles (UUV), and Intelligence Surveillance Reconnaissance (ISR) aircraft. Easily integrated into network centric systems, the CNS4 is an easy to use, turnkey, rugged network File Server that houses four Flash Storage Modules (FSMs) that provides quick off load of data. The FSMs can be easily removed from one CNS4 and installed into any other CNS4 providing full, seamless data transfer between one or more networks in separate locations (e.g. ground => vehicle => ground). In addition to the software-based FDE layer provided by the CNS4, the CNS4 provides a hardware-based Full Drive Encryption (FDE) layer to encrypt the drives within the FSMs. The hardware-based FDE layer is addressed in a separate evaluation.

The Curtiss-Wright product supports networking protocols including SSH, CIFS, NFS, FTP, HTTP, DHCP, SNMP, and iSCSI in addition to its RS-232 console port. The FDEEEcPP20E and FDEAAcPP20E Protection Profiles did not consider, nor did they include networking protocols as part of the security functional requirements, and as a result, did not include any requirements for addressing those protocols. Therefore, as per the FDEEEcPP20E and FDEAAcPP20E, the protocols have not been examined as part of the required assurance activities and consequently the evaluation can make no claims about the TOE's networking protocols.

It is suggested that a customer using the product consider the impact of utilizing remote administration via SSH across the network (rather than through the console) based upon their specific use case. The customer should factor into their risk management decision the environment in which TOE operates (dedicated, segregated, private network versus residing in a DMZ accessible to the Internet), and the value of data to be protected.

1.4.1 TOE Architecture

The TOE provides a software Full Drive Encryption solution that can accept Flash Storage Modules (FSMs) which contain data drives within. The software encryption is based on Linux Unified Key Setup (LUKS). The LUKS-based encryption is layered on the hardware encryption that was evaluated in a separate evaluation. This evaluation addresses the encryption of data in software-based partitions while the hardware evaluation focused on the raw disk.

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 CNS4 (SW 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 all partitions on the 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 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 fail, the TOE will not go into an operational mode.

1.4.2 TOE Documentation

CNS4 CSfC Common Airborne Recorder CSfC Encrypted Data Storage User Guide Part Number: DDOC0108-000-A2 **[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, 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:
    - Applicable NIAP Technical decisions: None

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 Compact Network Storage 4-Slot Software Encryption Layer 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 than 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:

- FCS_AFA_EXT.1: Authorization Factor Acquisition
- FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition
- FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
- FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
- FCS_CKM_EXT.6: Cryptographic Key Destruction Types
- FCS_KDF_EXT.1: Cryptographic Key Derivation
- FCS_KYC_EXT.1: Key Chaining (Initiator)
- FCS_KYC_EXT.2: Key Chaining (Recipient)
- FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- FCS_VAL_EXT.1: Validation
- FDP_DSK_EXT.1: Protection of Data on Disk
- FPT_KYP_EXT.1: Protection of Key and Key Material
- FPT_PWR_EXT.1: Power Saving States
- FPT_PWR_EXT.2: Timing of Power Saving States
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Trusted Update

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 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the FDEEEcPP20E/FDEAAcPP20E that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. 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 Compact Network Storage 4-Slot Software Encryption Layer TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_AFA_EXT.1: Authorization Factor Acquisition
	FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition
	FCS_CKM.1(c): Cryptographic Key Generation (Data Encryption Key)
	FCS_CKM.4(a): Cryptographic Key Destruction (Power Management)
	FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
	FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
	FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
	FCS_CKM_EXT.6: Cryptographic Key Destruction Types
	FCS_COP.1(a): Cryptographic Operation (Signature Verification)
	FCS_COP.1(b): Cryptographic operation (Hash Algorithm)
	FCS_COP.1(c): Cryptographic operation (Keyed Hash Algorithm)
	FCS_COP.1(d): Cryptographic operation (Key Wrapping)
	FCS_COP.1(f): Cryptographic operation (AES Data Encryption/Decryption)
	FCS_COP.1(g): Cryptographic operation (Key Encryption)
	FCS_KDF_EXT.1: Cryptographic Key Derivation
	FCS_KYC_EXT.1: Key Chaining (Initiator)
	FCS_KYC_EXT.2: Key Chaining (Recipient)
	FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
	FCS_VAL_EXT.1: Validation
FDP: User data protection	FDP_DSK_EXT.1: Protection of Data on Disk
FMT: Security management	FMT_MOF.1: Management of Functions Behavior
	FMT_SMF.1: Specification of Management Functions

	FMT_SMR.1: Security Roles
FPT: Protection of the TSF	FPT_KYP_EXT.1: Protection of Key and Key Material
	FPT_PWR_EXT.1: Power Saving States
	FPT_PWR_EXT.2: Timing of Power Saving States
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Trusted Update

Table 1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Authorization Factor Acquisition (FCS_AFA_EXT.1)

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***].

5.1.1.2 Timing of Authorization Factor Acquisition (FCS_AFA_EXT.2)

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

FCS_CKM.1(c).1

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***].

5.1.1.4 Cryptographic Key Destruction (Power Management) (FCS_CKM.4(a)) [1]

FCS_CKM.4(a).1

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

5.1.1.5 Cryptographic Key Destruction (Software TOE, 3rd Party Storage) (FCS_CKM.4(d))

FCS_CKM.4(d).1

Refinement: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [
- For volatile memory, the destruction shall be executed by a [
- single overwrite consisting of [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 [o logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeros]]] that meets the following: no standard.

[1] The FDEEEcPP20E version is used as it addresses the FDEAAcPP20E requirement by requiring key destruction as specified in FCS_CKM.4(d). In this ST, the FCS_CKM_EXT.6 requirement points to FCS_CKM.4(d).

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

FCS_CKM_EXT.4(a).1

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

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

FCS_CKM_EXT.4(b).1

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.

5.1.1.8 Cryptographic Key Destruction Types (FCS_CKM_EXT.6)

FCS_CKM_EXT.6.1

The TSF shall use [***FCS_CKM.4(d)***] key destruction methods.

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

FCS_COP.1(a).1

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

5.1.1.10 Cryptographic operation (Hash Algorithm) (FCS_COP.1(b))

FCS_COP.1(b).1

Refinement: The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-256, SHA-512***] that meet the following: ISO/IEC 10118-3:2004.

5.1.1.11 Cryptographic operation (Keyed Hash Algorithm) (FCS_COP.1(c))

FCS_COP.1(c).1

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

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

FCS_COP.1(f).1

The TSF shall perform data 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***].

5.1.1.13 Cryptographic operation (Key Encryption) (FCS_COP.1(g))

FCS_COP.1(g).1

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***].

5.1.1.14 Cryptographic Key Derivation (FCS_KDF_EXT.1)

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.

5.1.1.15 Key Chaining (Initiator) (FCS_KYC_EXT.1)

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.

FCS_KYC_EXT.1.2

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

5.1.1.16 Key Chaining (Recipient) (FCS_KYC_EXT.2)

FCS_KYC_EXT.2.1

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

FCS_KYC_EXT.2.2

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

5.1.1.17 Cryptographic Password Construct and Conditioning (FCS_PCC_EXT.1)

FCS_PCC_EXT.1.1

A password used by the TSF to generate a password authorization factor shall enable up to [***512***] 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-256***], with [***89,0001***] iterations, and output cryptographic key sizes [***256 bits***] that meet the following: NIST SP 800-132.

5.1.1.18 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with [***ISO/IEC 18031:2011***] using [***HMAC_DRBG (any)***]].

FCS_RBG_EXT.1.2

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

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

FCS_SNI_EXT.1.1

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

FCS_SNI_EXT.1.2

The TSF shall use [***no nonces***].

FCS_SNI_EXT.1.3

The TSF shall create IVs in the following manner [***- CBC: IVs shall be non-repeating and unpredictable***].

5.1.1.20 Validation (FCS_VAL_EXT.1)[2]

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***].

FCS_VAL_EXT.1.2

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

FCS_VAL_EXT.1.3

The TSF shall [***- block validation after [three attempts] of consecutive failed validation attempts - require power cycle/reset the TOE after [three attempts] of consecutive failed validation attempts***].

5.1.2 User data protection (FDP)

5.1.2.1 Protection of Data on Disk (FDP_DSK_EXT.1)

FDP_DSK_EXT.1.1

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

FDP_DSK_EXT.1.2

The TSF shall encrypt all protected data without user intervention.

5.1.3 Security management (FMT)

5.1.3.1 Management of Functions Behavior (FMT_MOF.1)

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 (FMT_SMF.1)

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 factors or set of authorization factors used,
d) change the DEK, as specified in FCS_CKM.1, when reprovisioning or when commanded,
e) erase the DEK, as specified in FCS_CKM.4(a), [selection:
f) initiate TOE firmware/software updates,
g) [***no other functions***]

5.1.3.3 Security Roles (FMT_SMR.1)

FMT_SMR.1.1

The TSF shall maintain the roles authorized users.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

[2] The FDEEEcPP20E version is used as it addresses the FDEAAcPP20E requirement by including the BEV in the first element and requiring validation of the BEV in the second element.

5.1.4 Protection of the TSF (FPT)

5.1.4.1 Protection of Key and Key Material (FPT_KYP_EXT.1)

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)***].

5.1.4.2 Power Saving States (FPT_PWR_EXT.1)

FPT_PWR_EXT.1.1
The TSF shall define the following Compliant power saving states: [***G3***].

5.1.4.3 Timing of Power Saving States (FPT_PWR_EXT.2)

FPT_PWR_EXT.2.1
For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur:
user-initiated request,
[***shutdown,***
request initiated by remote management system].

5.1.4.4 TSF Testing (FPT_TST_EXT.1)

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***].

5.1.4.5 Trusted Update (FPT_TUD_EXT.1)

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

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

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

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 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***].

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:

- FCS_AFA_EXT.1: The TOE supports password authorization factor, and the password may be between 8 and (up to) 512 characters in length and can be composed of all ASCII printable characters.
- 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 administrator must authenticate and make the encrypted partition available before the TOE will allow data to be read from or written to the drive.
- FCS_CKM.1(c): The TOE generates 256-bit DEKs using its SHA-256 HMAC_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 unpartitioned, drives or at the start of each partition).
- 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.
- FCS_CKM.4(d): The TOE has 4GB of 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. The TOE also clears keys when they are no longer in use by overwriting them with zeroes.

  Additionally, the TOE stores encrypted DEKs in a header for the encrypted drive partitions. The TOE clears these keys by through an internal call using the CRYPT_WIPE_RANDOM pattern, which draws random data from the TOE's HMAC_DRBG.
- 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 logs out, then the TOE will clear any in-use DEKs from kernel memory. Keys are also cleared upon removal of power.
- FCS_CKM_EXT.4(b): The TOE has no Compliant power saving states other than power on and off (G3).
- 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 OpenSSL library when verifying ECDSA P-521 w/ SHA-512 trusted update signatures.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(a) (Verify)	ECDSA P-521 w/ SHA-512 Verify	FIPS 186-4, ECDSA	1500
FCS_COP.1(b) (Hash)	SHA-512 Hashing	FIPS 180-4	4471

Table 3 OpenSSL Cryptographic Algorithms

The TOE uses its kernel cryptography when doing AES-256 CBC ESSIV:SHA-256 data encryption/decryption.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(b) (Hash)	SHA-256 Hashing	FIPS 180-4	4470
FCS_COP.1(f) (AES)	AES-256 CBC Encrypt/Decrypt	FIPS 197	5564

Table 4 kernel Cryptographic Algorithms

The TOE uses its libgcrypt library when doing key derivation and key management operations.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(b) (Hash)	SHA-256 Hashing	FIPS 180-4	4472
FCS_COP.1(d) (Keyed Hash)	HMAC-SHA-256	FIPS 198-1 & 180-4	3709
FCS_COP.1(f) (AES)	AES-256 CBC Encrypt/Decrypt	FIPS 197	5565
FCS_RBG_EXT.1 (Random)	SHA-256 HMAC_DRBG	SP 800-90A	2218

Table 5 libgcrypt Cryptographic Algorithms

- FCS_COP.1(a): The TOE utilizes ECDSA P-521 w/ SHA-512 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 (see FPT_TUD_EXT.1 below for the location) to verify the ECDSA signature against the received image. The verification uses SHA-512 and follows the FIPS 186-4 ECDSA format.

- FCS_COP.1(b): The TOE's kernel, libgcrypt, and OpenSSL libraries provide the SHA-256 and SHA-512 algorithms and use those algorithms as part of ESSIV:SHA-256 IV generation, PBKDFv2 password-based key derivation, and trusted update signature verification respectively.

- FCS_COP.1(c): The TOE implements HMAC-SHA-256 using 256-bit keys, the SHA-256 hash algorithm, a 512-bit block size, and an output MAC length of 256 bits.

- FCS_COP.1(f): The TOE uses an AES CBC kernel implementation dedicated to drive encryption/decryption. This implementation uses AES- 256 bit keys.

- FCS_COP.1(g): The TOE has a libgcrypt AES CBC implementation used for key managements operations (decryption of the encrypted DEKs). This implementation uses AES- 256 bit keys.

- FCS_KDF_EXT.1: The TOE uses 800-132 (PBKDFv2) with HMAC-SHA-256 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-sha256 of 44,521 iterations per second for a 256-bit key, and thus the TOE uses a count of ~89,000 iterations to achieve a 2 second delay.

- 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.

- FCS_PCC_EXT.1: The TOE allows passwords up to 512 characters in length, and the TOE allows uppercase/lowercase letters, numbers, and ASCII printable characters. The TOE will reject a password containing other characters. The TOE conditions passwords by combining them with a 256-bit salt using PBKDFv2.

- 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.

- FCS_SNI_EXT.1: The TOE generates its salts using its SHA-256 HMAC_DRBG. The TOE generates its AES-CBC IVs using ESSIV:SHA256. The TOE does not generate nonces nor tweaks (as the TOE doesn’t support AES-XTS).

- 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-256 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 detects more than three incorrect passwords, then the TOE will block all subsequent attempts to validate the operator’s password (and not even attempt to validate the password). The TOE clears its counter upon a reboot.

6.2 User data protection

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

- FDP_DSK_EXT.1: The TOE provides FDE that encrypts the entirety of each drive partition through AES-CBC block based encryption. The Admin Guide describes the TOE’s initialization process and setup for the SW-layer. The TOE maintains a separate, unencrypted, internal Flash chip to house its CentOS-based firmware that is beyond the FSM drive that the TOE encrypts. If the administrator configures the FSM drive for use as a raw block device, then the TOE encrypts the entire drive (with a small area reserved for the LUKS header). Otherwise, if the administrator chooses to partition the FSM drive, then the drive’s partition table and LUKS headers for each partition will be in plaintext, with all partition data encrypted.

6.3 Security management

The Security management function satisfies the following security functional requirements:

- FMT_MOF.1: The TOE claims no Compliant power saving states beyond power on and off. Only the authorized administrator can issue the shutdown command.
- 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. Instead, the TOE provides an administrator command that will erase the DEK (“**`sysconfig --wipe --force`**”) and a command to create a new partition (“**`sysconfig -s 0 --part 2 50% 50% --force`**”). 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 the TOE’s “Field Update” process, which consists of securely copying the new update image and signature file to the TOE and then executing the “`fupdate`” command, after which the TOE will detect the new update, verify the signature, and (if the signature verifies successfully) install the update. The TOE does not provide any manageable power-saving states.
- FMT_SMR.1 – The TOE 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:

- FPT_KYP_EXT.1: The TOE stores encrypted DEKs in the header of each drive partition.
- FPT_PWR_EXT.1/2: The TOE provides the Compliant power-saving state G3, mechanical off. . The TOE enters this state when the user shuts off the device, or when the remote administrator shuts down the device. The TOE must be fully rebooted from this state.
- 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.
  - OpenSSL - ECDSA sign/verify test
  - OpenSSL – SHA-512 hashing test
  - OpenSSL – integrity test
  - kernel – SHA-256 hashing test
  - kernel – AES-256 CBC encrypt/decrypt test
  - kernel – integrity test
  - libgcrypt – SHA hashing tests

- libgcrypt – HMAC-SHA tests
- libgcrypt – AES-256 CBC encrypt/decrypt test
- libgcrypt – SHA-256 HMAC_DRBG test
- libgcrypt – integrity test

The TOE also has a set of functional self tests. The information from these tests is stored within the temporary file system and can be viewed using the 'log' command.

- PBIT (Power-on Built In Test) – performs a basic check of critical components on the system (tests the physical RAM and tests data communication paths.
- CBIT (Continuous Built In Test) - checks sensors, SATA status and errors
- IBIT (Instantiated Built In Test) - performs a more in-depth status check of components such as all communication paths, individual I2C devices, the eNOVA hardware encryptors, voltages, and S.M.A.R.T. data from the FSM-C's.

- FPT_TUD_EXT.1: The TOE can display its current firmware version and has the ability to field update its software using signed updates. The TOE will verify the signature on a firmware upgrade (using its OpenSSL library in conjunction with the embedded /root/fupdate/cwdts_publickey.pem key to verify the ECDSA P-521 with SHA-512 signature) before installing it, and will reject any update with an invalid signature. The signature used for update validation is protected because it lives on the operating system that is stored on Flash belonging to the dedicated, physically separate computing system.

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	Strength of Key	When Key Destroyed
User Passphrase	Memory - transient	N/A	N/A	N/A	Immediately after use
Derived Key	Memory	N/A	The TOE uses 800-132 KDF in counter mode using HMAC-SHA-256 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	Memory and Partition Header	AES CBC Encrypted	Generated from approved DRBG	256 bits	When partition closed or when partition no longer encrypted

Table 6 Key Identification

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 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 data encryption engine itself is a Network Attached Storage (NAS) device, where all executable code of the data encryption engine executes within a dedicated processor, with its own dedicated Flash memory. While the TOE does not encrypt its internal dedicated Flash 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 Flash, the TOE only provides access to the FSM/drives once fully initialized and after receiving the administrator's password.

The TOE uses 3 crypto modules:

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