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

Version 0.4
March 26, 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 Hardware Encryption Layer provided by Curtiss-Wright Defense Solutions. The TOE is being evaluated as a hardware full drive encryption solution.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - 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 Hardware Encryption Layer (FDEEEcPP20E/FDEAAcPP20E) Security Target

ST Version – Version 0.4

ST Date – March 26, 2019

1.2 TOE Reference

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

The TOE provides hardware Full Drive Encryption of a removable drive.

1.4 TOE Description

The Curtiss-Wright Defense Solutions Compact Network Storage 4-Slot Hardware 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 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 the CNS4 and installed into any other providing full, seamless data transfer between one or more networks in separate locations (e.g. ground => vehicle => ground).

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 hardware Full Drive Encryption solution that can accept a Flash Storage Module (FSM) which contains a data drive within.

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 (HW Layer):

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

1.4.1.2.1 Cryptographic support

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

1.4.1.2.2 User data protection

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

1.4.1.2.3 Security management

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

1.4.1.2.4 Protection of the TSF

The TOE implements a number of features to protect itself to ensure the reliability and integrity of its security features. It protects key and key material, and includes functions to perform self-tests and software/firmware integrity checking so that it might detect when it is failing or may be corrupt. If any of the self-tests 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 Hardware 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)
- FDEEEcPP20: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 Hardware 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(b): Cryptographic key generation (Symmetric Keys)
	FCS_CKM.1(c): Cryptographic Key Generation (Data Encryption Key)
	FCS_CKM.4(a): Cryptographic Key Destruction (Power Management)
	FDEEEcPP20:FCS_CKM.4(b): Cryptographic Key Destruction (TOE-Controlled Hardware)
	FDEAAcPP20:FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage)
	FCS_CKM.4(e): Cryptographic Key Destruction (Key Cryptographic Erase)
	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)
	FDEEEcPP20: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_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_FUA_EXT.1: Firmware Update Authentication
	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 (Symmetric Keys) (FCS_CKM.1(b))

FCS_CKM.1(b).1

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) (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,***
- accept a DEK that is wrapped as specified in FCS_COP.1(d)] and specified cryptographic key sizes [***256 bits***].

5.1.1.5 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

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

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.

5.1.1.6 Cryptographic Key Destruction (TOE-Controlled Hardware) (FDEEEcPP20:FCS_CKM.4(b))

FDEEEcPP20:FCS_CKM.4(b).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 [***
  - ***removal of power to the memory]***
- ***For non-volatile memory [that does not employ a wear-leveling algorithm, the destruction shall be executed by a [***
  - ***[single] overwrite consisting of zeros followed by a read-verify,***
  - ***[single] overwrite consisting of ones followed by a read-verify,***
  - ***overwrite with a new value of a key of the same size followed by a read-verify,***
  - ***[single] overwrite consisting of [data from an RBG] followed by a read-verify***

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

] that meets the following: no standard.

5.1.1.7 Cryptographic Key Destruction (Software TOE, 3rd Party Storage) (FDEAAcPP20:FCS_CKM.4(d))

FDEAAcPP20: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 overwright consisting of [***
    - ***zeros***
    - ***ones***
  - ***- removal of power to the memory]***
- ***For non-volatile storage that consists of the invocation of an interface provided by***
- ***the underlying platform that [logically addresses the storage location of the key and performs a [[***
  - ***[single] overwrite consisting of zeros followed by a read-verify,***
  - ***[single] overwrite consisting of ones followed by a read-verify,***
  - ***overwrite with a new value of a key of the same size followed by a read-verify,***
  - ***[single] overwrite consisting of [data from an RBG] followed by a read-verify***

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

] that meets the following: no standard.

5.1.1.8 Cryptographic Key Destruction (Key Cryptographic Erase) (FCS_CKM.4(e))

FCS_CKM.4(e).1

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

5.1.1.9 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.

5.1.1.10 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.

5.1.1.11 Cryptographic Key Destruction Types (FDEEEcPP20:FCS_CKM_EXT.6)

FDEEEcPP20:FCS_CKM_EXT.6.1

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

5.1.1.12 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-384]; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes***].

5.1.1.13 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-384***] that meet the following: ISO/IEC 10118-3:2004.

5.1.1.14 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 used in [HMAC]***] that meet the following: [***ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'***].

5.1.1.15 Cryptographic operation (Key Wrapping) (FCS_COP.1(d))

FCS_COP.1(d).1

Refinement: The TSF shall perform key wrapping in accordance with a specified cryptographic algorithm AES in the following modes [***KW***] and the cryptographic key size [***256 bits***] that meet the following: AES as specified in ISO/IEC 18033-3, [***NIST SP 800-38F***].

5.1.1.16 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***].

5.1.1.17 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.

5.1.1.18 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***].

5.1.1.19 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 wrapping as specified in FCS_COP.1(d)***] while maintaining an effective strength of [***256 bits***] for symmetric keys and an effective strength of [***not applicable***] for asymmetric keys.

5.1.1.20 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 [***64***] characters in the set of upper case characters, lower case characters, numbers, and [***no special characters***] and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[***SHA-256***], with [***1000***] iterations, and output cryptographic key sizes [***256 bits***] that meet the following: NIST SP 800-132.

5.1.1.21 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 [***NIST SP 800-90A***] using [***HMAC_DRBG (any), CTR_DRBG (AES)***]].

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.

5.1.1.22 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***].

5.1.1.23 Validation (FCS_VAL_EXT.1)[2]

FCS_VAL_EXT.1.1

The TSF shall perform validation of the BEV using the following method(s): [***- key wrap as specified in FCS_COP.1(d)***]

FCS_VAL_EXT.1.2

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

FCS_VAL_EXT.1.3

The TSF shall [***- require power cycle/reset the TOE after [three] 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),
f) initiate TOE firmware/software updates,
g) [***import a wrapped DEK***]

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 Update Authentication (FPT_FUA_EXT.1)

FPT_FUA_EXT.1.1

The TSF shall authenticate the source of the firmware update using the digital signature algorithm specified in FCS_COP.1(a) using the RTU that contains [***the public key***].

FPT_FUA_EXT.1.2

The TSF shall only allow installation of update if the digital signature has been successfully verified as specified in FCS_COP.1(a).

FPT_FUA_EXT.1.3

The TSF shall only allow modification of the existing firmware after the successful validation of the digital signature, using a mechanism as described in FPT_TUD_EXT.1.2.

FPT_FUA_EXT.1.4

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

5.1.4.2 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),***
- ***only store plaintext keys that meet any one of the following criteria [***
  - ***The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1/2[3]]***

]

5.1.4.3 Power Saving States (FPT_PWR_EXT.1)

FPT_PWR_EXT.1.1

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

5.1.4.4 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,
[***no other conditions***].

5.1.4.5 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***].

[3] The FDEAAcPP20E has FCS_KYC_EXT.1 and the FDEEEcPP20E has FCS_KYC_EXT.2 for this selection so both are included here.

5.1.4.6 Trusted Update (FPT_TUD_EXT.1)[4]

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

5.2 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.

[4] The FDEEEcPP20E version is used as it addresses the FDEAAcPP20E requirement by including FPT_FUA_EXT.1 in the third element it requires a digital signature as implemented in FCS_COP.1(a).

ADV_FSP.1.2e

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

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

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

AGD_OPE.1.2c

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

AGD_OPE.1.3c

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

AGD_OPE.1.4c

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

AGD_OPE.1.5c

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

AGD_OPE.1.6c

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

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

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

AGD_PRE.1.2c

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

AGD_PRE.1.1e

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

AGD_PRE.1.2e

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

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

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

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

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

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Security Target (ASE)

5.2.4.1 Cryptographic operation (Hash Algorithm) (ASE_TSS.1(c))

ASE_TSS.1(c).1

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

ASE_TSS.1(c).1: Section 7 provides the TOE's Key Management Description, the separate Entropy Documentation and Analysis document provides the TOE's Entropy Essay, and the TOE includes the following 3rd party hardware components.

Component	Version/Part Number
Microcontroller	NXP (Phillips) ARM7 Processor P/N LPC2368FBD100
Maxim Security Chip	Maxim Integrated DS3645, Rev A4
Entropy Chip	Microchip ATECC508A
HW Encryption Chip	Enova X-Wall MX-256C version 1.0
Secondary Storage Chip	Microchip 24LC512

Table 3 3rd Party Hardware Components

5.2.5 Tests (ATE)

5.2.5.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability assessment (AVA)

5.2.6.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d
The developer shall provide the TOE for testing.

AVA_VAN.1.1c
The TOE shall be suitable for testing.

AVA_VAN.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2e
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

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

6.1 Cryptographic support

The Cryptographic support function satisfies the following security functional requirements:

- FCS_AFA_EXT.1: The TOE supports a password authorization factor, and the password may be up to 64 characters (bytes) in length and can be composed of uppercase and lowercase letters and numbers.
- 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 authenticate before the TOE will allow data to be read from or written to the drive.
- FCS_CKM.1(b): The TOE generates 256-bit DEKs and KEKs using its SHA-256 HMAC_DRBG. The TOE stores these keys in its hardware keystore (dedicated, separate memory).
- FCS_CKM.1(c): The TOE can both generate 256-bit DEKs onboard using its HMAC_DRBG as well as accept injection of externally supplied 256-bit DEKs. The TOE uses its cm_key interface to load the externally generated DEKs. The TOE AES-KW unwraps injected DEKs using the 256-bit KEK stored internally.
- FCS_CKM.4(a): When the TOE powers off (as the TOE has no other power states other than on and off), all values in normal SRAM memory drain to a zero state (as opposed to the BB-SRAM—see below).
- FDEEEcPP20:FCS_CKM.4(b)/FDEAAcPP20:FCS_CKM.4(d): The TOE includes a working memory, a separate battery-backed SRAM, a entropy chip with internal EEPROM, and a secondary storage EEPROM chip. The TOE includes working memory RAM as part of its microcontroller, and this serves as the working memory in which the TOE stores the AES-KW key (derived from the password and salt) and the most recently used PSK or KEK (as part of a DEK import operation), and working copies of loaded DEKs (unwrapped during authentication). The TOE destroys these values when power is removed.

  The TOE's possesses a battery-backed SRAM (BB-SRAM) in which the TOE stores both the encrypted DEKs (the TOE can store copies of up to 4 encrypted DEKs for the administrator and up to four user accounts). The TOE utilizes its entropy chip to store the PSK. The TOE creates encrypted DEKs within its internal persistent storage (either the BB-SRAM or the secondary EEPROM, see below) in response to one of two administrator commands. First, the administrator can instruct the TOE to generate a random DEK, or second, the administrator can import a new DEK. The TOE stores the new DEK wrapped with its AES-KW key. Subsequently, the administrator can overwrite Individual DEKs by loading a new DEK or requesting a newly generated DEK and specifying the key ID containing of key to be overwritten. When the administrator does this, the TOE overwrites the previous value with the new value (reusing the exact same location). Additionally, any operator may request that the TOE zeroize its keys, in which case, the TOE will overwrite its BB-SRAM and entropy chip storage with (in this order) random data from the RBG, all zeros, and then all ones (in accordance with the DoD requirements for a standard erase).

  The TOE also includes a secondary EEPROM storage chip, which the administrator can initialize the TOE to use, in lieu of the BB-SRAM, for persistent key storage. The TOE's follows the same behavior (in terms

of the number of keys, their generation, their protection with AES-KW, and key clearing methods) whether using its BB-SRAM or EEPROM for storage.

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

The TOE's second EEPROM chip is a 512Kb EEPROM that is 128-byte page aligned, which allows single byte read/write operations.

- FCS_CKM.4(e): The TOE can, in effect, cryptographically erase all stored DEKs (in addition to directly clearing the encrypted DEKs) by destroying the salt value associated with the operator's password. Once destroyed, no one can derive the AES-KW key needed to decrypt encrypted DEKs without the 256-bit salt value.
- FCS_CKM_EXT.4(a): The TOE considers keys that the operator has explicitly requested to be deleted as no longer necessary, and if reset, the TOE deems all keys other than the PSK as no longer needed. For those keys, the TOE will erase its hardware keystore (a battery-backed SRAM circuit or EEPROM depending on how the administrator initialized the TOE).
- FCS_CKM_EXT.4(b): The TOE has the Compliant power saving state of G3 (Mechanical Off).
- FDEEEcPP20:FCS_CKM_EXT.6: The TOE clears its keys in accordance with FCS_CKM.4(b).
- FCS_COP.1: The TOE performs cryptographic algorithms in accordance with the following NIST standards and has received the following CAVP algorithm certificates. Note that the TOE includes the Enova X-Wall AES-CBC hardware chip and the Microchip ATECC508A entropy chip. Note that Microchip's DRBG CAVP certificate (#532) tests their DRBG as instantiated in the 108A version of the part which shares the same silicon and DRBG ROM code as the 508A version.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(a) (Verify)	ECDSA P-384 w/SHA-384 Verify	FIPS 186-4, ECDSA	1551
FCS_COP.1(b) (Hash)	SHA-256/384 Hashing	FIPS 180-4	4590
FCS_COP.1(c) (Keyed Hash)	HMAC-SHA-256	FIPS 198-1 & 180-4	3815
FCS_COP.1(d) (Key Wrap)	AES-256 KW	FIPS 197, SP 800-38F	5767
FCS_COP.1(f) (AES)	AES-256 CBC Encrypt/Decrypt	FIPS 197	250
FCS_RBG_EXT.1 (Random)	AES-128 CTR_DRBG	SP 800-90A	532
FCS_RBG_EXT.1 (Random)	SHA-256 HMAC_DRBG	SP 800-90A	2362

Table 4 Cryptographic Algorithms

- FCS_COP.1(a): The TOE utilizes ECDSA P-384 with SHA-384 signatures to verify the authenticity of firmware updates. Upon receiving a candidate update and the accompanying signature file, the TOE uses an embedded public key to verify the ECDSA signature accompanying the received image. The verification uses SHA-384 and follows the FIPS 186-4 ECDSA format.
- FCS_COP.1(b): The TOE implements the SHA-256 and SHA-384 algorithms and uses the SHA-256 algorithm as part of PBKDFv2 key derivation and also as part of its HMAC_DRBG. The TOE uses SHA-384 hashing when verifying trusted update ECDSA P-384 signatures.
- FCS_COP.1(c): The TOE implements HMAC-SHA-256 using a 256-bit key, the SHA-256 hash algorithm, a 512-bit block size, and an output MAC length of 256-bits.
- FCS_COP.1(d): The TOE uses AES-KW (compliant with NIST SP 800-38F) to encrypt the DEKs stored in battery-backed memory and to unwrap DEKs injected by an administrator.

- FCS_COP.1(f): The TOE possesses an AES CBC implementation dedicated to drive encryption/decryption. The TOE's implementation exclusively uses 256-bit keys.
- FCS_KDF_EXT.1: The TOE uses 800-132 PBKDF in counter mode using SHA-256 and 1000 iterations and a 256-bit salt to transform the operator's password into a key for wrapping/unwrapping.
- FCS_KYC_EXT.1/2: The TOE uses PBKDFv2 to transform the operator's password into a 256-bit BEV, and then uses that BEV to AES-KW unwrap the DEKs stored in the TOE hardware key store. The AES-KW unwrap operation will verify whether the operator supplied the correct password. If the operator supplied the correct password, then the TOE will have access to the DEK values (in memory).
- FCS_PCC_EXT.1: The TOE allows passwords up to 64-bytes (characters in length), and the TOE checks to only allow uppercase/lowercase letters and numbers. The TOE will reject a password containing other characters. The TOE conditions passwords by combining them with a 256-bit salt using PBKDFv2.
- FCS_RBG_EXT.1: The TOE includes a firmware SHA-256 HMAC_DRBG that it seeds with at least 256-bits of entropy from a hardware-based noise source. The entropy source includes a hardware AES-128 CTR_DRBG that is seeded with 256-bits of hardware noise entropy.
- FCS_SNI_EXT.1: The TOE generates its salts (each account, whether admin or user, has a 256-bit salt used during PBKDFv2 derivation) and AES-CBC IVs using its SHA-256 HMAC_DRBG. The TOE does not generate nonces nor tweaks (as the TOE doesn't support AES-XTS).
- FCS_VAL_EXT.1: A password is required when the machine is power cycled. The TOE validates the operator's password by attempting an AES-KW decrypt/unwrap operation. If the AES-KW decryption operation fails, then the TOE treats this as an invalid login and increments its failed login attempts counter. If the counter has reached three, the TOE enters a hard error requiring a power cycle. The TOE resets its counter upon a reset or upon a successful authentication.

6.2 User data protection

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

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

6.3 Security management

The Security management function satisfies the following security functional requirements:

- 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 allows an administrator can change a DEK, overwrite a DEK, cryptographically erase all DEKs, import a wrapped DEK using the cm_key command. The TOE supports changing of the authorization factors (the administrator can zeroize the TOE and then reinitialize the TOE to change the associated passwords) using the cm_key command to zeroize and the cmlogin command to set a new password.
- 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_FUA_EXT.1: The TOE uses an internal ECDSA P-384 public key (hardcoded within the TOE's existing microprocessor firmware image, stored within the microprocessor) to verify new firmware images before writing the firmware to the TOE's internal storage.
- FPT_KYP_EXT.1: The TOE store keys in its battery-backed SRAM chip (part of its hardware keystore), which acts like non-volatile memory. The TOE stores all keys in encrypted form (encrypted with an AES-KW key derived from the operator's password plus an internal salt) with the exception of the Pre-Shared Key (PSK). The PSK exists to protect the export of a transient KEK from the TOE. After export, the administrator can then use the KEK to wrap an DEK for import into the TOE, and the transient KEK is not kept across a reset of the TOE. In this way, the PSK does not participate in the key chain protecting DEKs, but instead, only provides an extra layer of transient protection during DEK injection.
- 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 administrator issues the shutdown command. 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.
  - ECDSA verify test
  - SHA-256/384 hashing tests
  - HMAC-SHA-256 hashing test
  - AES CBC encryption/decryption test
  - SHA-256 HMAC_DRBG tests (including SP 800-90A section 11.3 health tests)
  - integrity test
- FPT_TUD_EXT.1: The TOE allows field updates to the HW-layer's firmware that have been signed and delivered by Curtiss Wright. The TOE will verify the ECDSA P-384 with SHA-384 signature of the update image, and if valid, the TOE will update its firmware. The signature used to validate the image is store on the firmware and is not modifiable.

7. Key Management Description

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

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

Table 5 Key Identification

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

The HW component receives the DEK from the microcontroller and then encrypts/decrypts data written to/read from the encrypted partition/drive. The TOE encrypts the keys its internal, dedicated battery-backed SRAM memory or secondary EEPROM chip, 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 BB-SRAM, the TOE only provides access to the FSM/drives once fully initialized and after receiving the administrator's password.

The TOE uses the following cryptographic implementations:

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