Galleon Embedded Computing XSR and G1 Hardware Encryption Layer Security Target

Version 1.5
July 14, 2022

Prepared for:

Galleon Embedded Computing AS

Hovfaret 10
N-0275 OSLO, Norway

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 Galleon Embedded Computing XSR and G1 Hardware Encryption Layer provided by Galleon Embedded Computing. 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 – Galleon Embedded Computing XSR and G1 Hardware Encryption Layer Security Target

ST Version – Version 1.5

ST Date – July 14, 2022

1.2 TOE Reference

TOE Identification – Galleon Embedded Computing XSR and G1 Hardware Encryption Layer

TOE Developer – Galleon Embedded Computing

Evaluation Sponsor – Galleon Embedded Computing

1.3 TOE Overview

The Target of Evaluation (TOE) is the Galleon Embedded Computing Encryption Module (EM) running firmware version 4.0.11. The Encryption Module as integrated into Galleon's XSR and G1 products provides their Hardware Encryption Layer, encrypting both internal, non-removable drives as well as a removable drive.

In adherence with NIAP Technical Decision TD0606, the TOE's evaluated configuration requires that the administrator configure the TOE during provisioning to enable local management (serial terminal over RS-232), unplug the ethernet interface, and disable remote management (TLS and SSH connections). Please see Section 2.2 of the TOE' Administrative Guide which details the evaluated configuration ("locally managed with remote management disabled").

1.4 TOE Description

The TOE comprises a dedicated hardware solution embedded into Galleon's XSR and G1 products. The TOE (known as the Hardware Encryption Layer or as the Encryption Module) takes the form of either a physically separate card (within the XSR) or directly integrated into the mainboard (of the G1).

The Encryption Module sits between the XSR and G1's mSATA connectors and the mSATA drives themselves (which includes an RDM, internal SSDs, and the non-removable mSATA[only present in the XSR]), providing transparent hardware-based Full Disk Encryption (FDE) of those drives.

The XSR and G1 products into which the TOE integrates can act in multiple different capacities (Network Attached Storage [NAS], data recorder, general server, etc.) and allow for encryption of the Removable Data Module (RDM) attached to the system. The Encryption Module within the XSR model supports encryption of one RDM (at a time), up to 4 internal SSDs, and its internal, non-removable mSATA SSD. Within the G1 model, the Encryption Module supports encryption of one RDM (at a time) and up to 2 internal SSDs. The Encryption Module securely encrypts all user data stored within.

In addition to the hardware-based FDE layer, the XSR and G1 products also possess a software-based Full Drive Encryption (FDE) layer; however, this software-based FDE layer is addressed in a separate evaluation.

1.4.1 TOE Architecture

The TOE provides a hardware Full Drive Encryption solution that can accept Removable Data Modules (RDMs) which contain data drives 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 the TOE (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 entirety of each drive (so that no plaintext exists) and does so without user intervention.

1.4.1.2.3 Security management

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

1.4.1.2.4 Protection of the TSF

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

1.4.2 TOE Documentation

Galleon Encryption Module Usage Guidelines Certifiable Encryption (Rev 1.0.6, July 14, 2022) **[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:

TD No.	Applied?	Rationale
FDEEEcPP20E:TD0606	Yes	The TOE is a NAS
FDEAAcPP20E:TD0606	Yes	The TOE is a NAS
FDEEEcPP20E:TD0464	Yes	FPT_PWR_EXT.1 claimed, SFR updated
FDEEEcPP20E:TD0460	Yes	FPT_PWR_EXT.1 claimed, AGD affected
FDEEEcPP20E:TD0458	Yes	FPT_KYP_EXT.1 claimed, TSS includes required info
FDEAAcPP20E:TD0458	Yes	FPT_KYP_EXT.1 claimed, TSS includes required info

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 Galleon Embedded Computing XSR and G1 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:

- FDEAAcPP20E:FCS_AFA_EXT.1: Authorization Factor Acquisition
- FDEAAcPP20E:FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition
- FDEAAcPP20E:FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
- FDEEEcPP20E:FCS_CKM_EXT.4(a): Cryptographic Key and Key Material Destruction (Destruction Timing)
- FDEAAcPP20E:FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
- FDEEEcPP20E:FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)
- FDEEEcPP20E:FCS_CKM_EXT.6: Cryptographic Key Destruction Types
- FDEAAcPP20E:FCS_KDF_EXT.1: Cryptographic Key Derivation
- FDEEEcPP20E:FCS_KDF_EXT.1: Cryptographic Key Derivation
- FDEAAcPP20E:FCS_KYC_EXT.1: Key Chaining (Initiator)
- FDEEEcPP20E:FCS_KYC_EXT.2: Key Chaining (Recipient)
- FDEAAcPP20E:FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning
- FDEAAcPP20E:FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FDEEEcPP20E:FCS_RBG_EXT.1: Random Bit Generation
- FDEAAcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- FDEEEcPP20E:FCS_SNI_EXT.1: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)
- FDEEEcPP20E:FCS_VAL_EXT.1: Validation
- FDEEEcPP20E:FDP_DSK_EXT.1: Protection of Data on Disk
- FDEEEcPP20E:FPT_FUA_EXT.1: Firmware Update Authentication
- FDEAAcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material
- FDEEEcPP20E:FPT_KYP_EXT.1: Protection of Key and Key Material
- FDEAAcPP20E:FPT_PWR_EXT.1: Power Saving States
- FDEEEcPP20E:FPT_PWR_EXT.1: Power Saving States
- FDEAAcPP20E:FPT_PWR_EXT.2: Timing of Power Saving States
- FDEEEcPP20E:FPT_PWR_EXT.2: Timing of Power Saving States
- FDEAAcPP20E:FPT_TST_EXT.1: TSF Testing
- FDEEEcPP20E:FPT_TST_EXT.1: TSF Testing
- FDEAAcPP20E:FPT_TUD_EXT.1: Trusted Update
- FDEEEcPP20E:FPT_TUD_EXT.1: Trusted Update

5. Security Requirements

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

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

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

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Galleon Embedded Computing XSR and G1 Hardware Encryption Layer TOE.

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

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

Table 1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Authorization Factor Acquisition (FDEAAcPP20E:FCS_AFA_EXT.1)

FDEAAcPP20E:FCS_AFA_EXT.1.1

The TSF shall accept the following authorization factors: [***- a submask derived from a password authorization factor conditioned as defined in FCS_PCC_EXT.1***].

5.1.1.2 Timing of Authorization Factor Acquisition (FDEAAcPP20E:FCS_AFA_EXT.2)

FDEAAcPP20E:FCS_AFA_EXT.2.1

The TSF shall reacquire the authorization factor(s) specified in FCS_AFA_EXT.1 upon transition from any Compliant power saving state specified in FPT_PWR_EXT.1 prior to permitting access to plaintext data.

5.1.1.3 Cryptographic Key Generation (Data Encryption Key) (FDEEEcPP20E:FCS_CKM.1(c))

FDEEEcPP20E:FCS_CKM.1.1(c)

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

5.1.1.4 Cryptographic Key Destruction (Power Management) (FDEAAcPP20E:FCS_CKM.4(a))

FDEAAcPP20E:FCS_CKM.4.1(a)

Refinement: The TSF shall [***erase***] cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1 that meets the following: a key destruction method specified in FCS_CKM.4(d).

5.1.1.5 Cryptographic Key Destruction (Power Management) (FDEEEcPP20E:FCS_CKM.4(a))

FDEEEcPP20E:FCS_CKM.4.1(a)

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

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

FDEEEcPP20:FCS_CKM.4.1(b)

Refinement: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- ***For volatile memory, the destruction shall be executed by a [***
  - ***[single overwrite consisting of [***
    - ***zeros]***
  - ***- removal of power to the memory]***
- ***For non-volatile memory [that employs a wear-leveling algorithm, the destruction shall be executed by a [***
  - ***block erase]***

]

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

FDEAAcPP20:FCS_CKM.4.1(d)

Refinement: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- ***For volatile memory, the destruction shall be executed by a [***
  - ***[single 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 [logically addresses the storage location of the key and performs a***
  - ***[single] overwrite consisting of [***
    - ***zeros]***

] that meets the following: no standard.

5.1.1.8 Cryptographic Key Destruction (Key Cryptographic Erase) (FDEEEcPP20E:FCS_CKM.4(e))

FDEEEcPP20E:FCS_CKM.4.1(e)

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

5.1.1.9 Cryptographic Key and Key Material Destruction (Destruction Timing) (FDEAAcPP20E:FCS_CKM_EXT.4(a))

FDEAAcPP20E:FCS_CKM_EXT.4.1(a)

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

5.1.1.10 Cryptographic Key and Key Material Destruction (Destruction Timing) (FDEEEcPP20E:FCS_CKM_EXT.4(a))

FDEEEcPP20E:FCS_CKM_EXT.4.1(a)

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

5.1.1.11 Cryptographic Key and Key Material Destruction (Power Management) (FDEAAcPP20E:FCS_CKM_EXT.4(b))

FDEAAcPP20E:FCS_CKM_EXT.4.1(b)

Refinement: The TSF shall destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1.

5.1.1.12 Cryptographic Key and Key Material Destruction (Power Management) (FDEEEcPP20E:FCS_CKM_EXT.4(b))

FDEEEcPP20E:FCS_CKM_EXT.4.1(b)

The TSF shall destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1.

5.1.1.13 Cryptographic Key Destruction Types (FDEEEcPP20E:FCS_CKM_EXT.6)

FDEEEcPP20E:FCS_CKM_EXT.6.1

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

5.1.1.14 Cryptographic Operation (Signature Verification) (FDEAAcPP20E:FCS_COP.1(a))

FDEAAcPP20E:FCS_COP.1.1(a)

Refinement: The TSF shall perform cryptographic signature services (verification) in accordance with a [***Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater***] that meet the following:[

- ***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***].

5.1.1.15 Cryptographic Operation (Signature Verification) (FDEEEcPP20E:FCS_COP.1(a))

FDEEEcPP20E:FCS_COP.1.1(a)

Refinement: The TSF shall perform cryptographic signature services (verification) in accordance with a [***Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater***] that meet the following:[

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

5.1.1.16 Cryptographic operation (Hash Algorithm) (FDEAAcPP20E:FCS_COP.1(b))

FDEAAcPP20E:FCS_COP.1.1(b)

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

5.1.1.17 Cryptographic Operation (Hash Algorithm) (FDEEEcPP20E:FCS_COP.1(b))

FDEEEcPP20E:FCS_COP.1.1(b)

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

5.1.1.18 Cryptographic operation (Keyed Hash Algorithm) (FDEAAcPP20E:FCS_COP.1(c))

FDEAAcPP20E:FCS_COP.1.1(c)

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

5.1.1.19 Cryptographic Operation (Message Authentication) (FDEEEcPP20E:FCS_COP.1(c))

FDEEEcPP20E:FCS_COP.1.1(c)

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

5.1.1.20 Cryptographic operation (Key Wrapping) (FDEAAcPP20E:FCS_COP.1(d))

FDEAAcPP20E:FCS_COP.1.1(d)

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

5.1.1.21 Cryptographic Operation (Key Wrapping) (FDEEEcPP20E:FCS_COP.1(d))

FDEEEcPP20E:FCS_COP.1.1(d)

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

5.1.1.22 Cryptographic operation (AES Data Encryption/Decryption) (FDEAAcPP20E:FCS_COP.1(f))

FDEAAcPP20E:FCS_COP.1.1(f)

Refinement: 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***].

5.1.1.23 Cryptographic Operation (AES Data Encryption/Decryption) (FDEEEcPP20E:FCS_COP.1(f))

FDEEEcPP20E:FCS_COP.1.1(f)

Refinement: The TSF shall perform data encryption and decryption in accordance with a specified cryptographic algorithm AES used in [***XTS***] mode and cryptographic key sizes [***256 bits***] that meet the following: AES as specified in ISO /IEC 18033-3, [***XTS as specified in IEEE 1619***].

5.1.1.24 Cryptographic Key Derivation (FDEAAcPP20E:FCS_KDF_EXT.1)

FDEAAcPP20E:FCS_KDF_EXT.1.1

The TSF shall accept [***a conditioned password submask***] to derive an intermediate key, as defined in [***NIST SP 800-132***] , using the keyed-hash functions specified in FCS_COP.1(c), such that the output is at least of equivalent security strength (in number of bits) to the BEV.

5.1.1.25 Cryptographic Key Derivation (FDEEEcPP20E:FCS_KDF_EXT.1)

FDEEEcPP20E:FCS_KDF_EXT.1.1

The TSF shall accept [***a conditioned password submask***] to derive an intermediate key, as defined in [***NIST SP 800-132***] , using the keyed-hash functions specified in FCS_COP.1(c), such that the output is at least of equivalent security strength (in number of bits) to the BEV.

5.1.1.26 Key Chaining (Initiator) (FDEAAcPP20E:FCS_KYC_EXT.1)

FDEAAcPP20E:FCS_KYC_EXT.1.1

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

FDEAAcPP20E:FCS_KYC_EXT.1.2

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

5.1.1.27 Key Chaining (Recipient) (FDEEEcPP20E:FCS_KYC_EXT.2)

FDEEEcPP20E:FCS_KYC_EXT.2.1

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

FDEEEcPP20E:FCS_KYC_EXT.2.2

The TSF shall maintain a chain of intermediary keys originating from the BEV to the DEK using the following method(s): [***- key wrapping as specified in FCS_COP.1(d)***] while maintaining an effective strength of [***256 bits***] for symmetric keys and an effective strength of [***not applicable***] for asymmetric keys.

5.1.1.28 Cryptographic Password Construct and Conditioning (FDEAAcPP20E:FCS_PCC_EXT.1)

FDEAAcPP20E:FCS_PCC_EXT.1.1

A password used by the TSF to generate a password authorization factor shall enable up to [***256***] characters in the set of upper case characters, lower case characters, numbers, and [***" ", "!", "", "#", "$", "%", "&", "'", "(", ")", "*", "+", ",", "-", ":", ";", "<", "=", "@", "[", "\", "]", "`", "{", "|", "}"***] and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[***SHA-384***], with [***10,000***] iterations, and output cryptographic key sizes [***256 bits***] that meet the following: NIST SP 800-132.

5.1.1.29 Extended: Cryptographic Operation (Random Bit Generation) (FDEAAcPP20E:FCS_RBG_EXT.1)

FDEAAcPP20E:FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with [***/NIST SP 800-90A/***] using [***CTR_DRBG (AES)***].

FDEAAcPP20E:FCS_RBG_EXT.1.2

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

5.1.1.30 Random Bit Generation (FDEEEcPP20E:FCS_RBG_EXT.1)

FDEEEcPP20E:FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with [***/NIST SP 800-90A/***] using [***CTR_DRBG (AES)***].

FDEEEcPP20E:FCS_RBG_EXT.1.2

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

5.1.1.31 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FDEAAcPP20E:FCS_SNI_EXT.1)

FDEAAcPP20E:FCS_SNI_EXT.1.1

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

FDEAAcPP20E:FCS_SNI_EXT.1.2
The TSF shall use [***no nonces***].
FDEAAcPP20E:FCS_SNI_EXT.1.3
The TSF shall create IVs in the following manner [
- CBC: IVs shall be non-repeating and unpredictable,
- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,].

5.1.1.32 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FDEEEcPP20E:FCS_SNI_EXT.1)

FDEEEcPP20E:FCS_SNI_EXT.1.1
The TSF shall [***use salts that are generated by a [DRBG as specified in FCS_RBG_EXT.1]***].
FDEEEcPP20E:FCS_SNI_EXT.1.2
The TSF shall use [***no nonces***].
FDEEEcPP20E:FCS_SNI_EXT.1.3
The TSF shall create IVs in the following manner [
- CBC: IVs shall be non-repeating and unpredictable,
- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,].

5.1.1.33 Validation (FDEEEcPP20E:FCS_VAL_EXT.1)

FDEEEcPP20E:FCS_VAL_EXT.1.1
The TSF shall perform validation of the BEV using the following method(s): [***- decrypt a known value using the [BEV] as specified in FCS_COP.1(f) and compare it against a stored known value***]
FDEEEcPP20E:FCS_VAL_EXT.1.2
The TSF shall require the validation of the BEV prior to allowing access to TSF data after exiting a Compliant power saving state.
FDEEEcPP20E:FCS_VAL_EXT.1.3
The TSF shall [***- require power cycle/reset the TOE after [5] consecutive failed validation attempts***].

5.1.2 User data protection (FDP)

5.1.2.1 Protection of Data on Disk (FDEEEcPP20E:FDP_DSK_EXT.1)

FDEEEcPP20E:FDP_DSK_EXT.1.1
The TSF shall perform Full Drive Encryption in accordance with FCS_COP.1(f), such that the drive contains no plaintext protected data.
FDEEEcPP20E:FDP_DSK_EXT.1.2
The TSF shall encrypt all protected data without user intervention.

5.1.3 Security management (FMT)

5.1.3.1 Management of Functions Behavior (FDEAAcPP20E:FMT_MOF.1)

FDEAAcPP20E:FMT_MOF.1.1
The TSF shall restrict the ability to [modify the behaviour of] the functions [use of Compliant power saving state] to [authorized users].

5.1.3.2 Specification of Management Functions (FDEAAcPP20E:FMT_SMF.1)

FDEAAcPP20E:FMT_SMF.1.1
The TSF shall be capable of performing the following management functions: [

a) forwarding requests to change the DEK to the EE,
b) forwarding requests to cryptographically erase the DEK to the EE,
c) allowing authorized users to change authorization factors or set of authorization factors used,
d) initiate TOE firmware/software updates,
e) [***no other functions***]

5.1.3.3 Specification of Management Functions (FDEEEcPP20E:FMT_SMF.1)

FDEEEcPP20E:FMT_SMF.1.1
The TSF shall be capable of performing the following management functions:
a) change the DEK, as specified in FCS_CKM.1, when reprovisioning or when commanded,
b) erase the DEK, as specified in FCS_CKM.4(a),
c) initiate TOE firmware/software updates,
d) [***no other functions***].

5.1.3.4 Security Roles (FDEAAcPP20E:FMT_SMR.1)

FDEAAcPP20E:FMT_SMR.1.1
The TSF shall maintain the roles [authorized user].

FDEAAcPP20E:FMT_SMR.1.2
The TSF shall be able to associate users with roles.

5.1.4 Protection of the TSF (FPT)

5.1.4.1 Firmware Update Authentication (FDEEEcPP20E:FPT_FUA_EXT.1)

FDEEEcPP20E:FPT_FUA_EXT.1.1
The TSF shall authenticate the source of the firmware update using the digital signature algorithm specified in FCS_COP.1(a) using the RTU that contains [***the public key***].
FDEEEcPP20E:FPT_FUA_EXT.1.2
The TSF shall only allow installation of update if the digital signature has been successfully verified as specified in FCS_COP.1(a).
FDEEEcPP20E:FPT_FUA_EXT.1.3
The TSF shall only allow modification of the existing firmware after the successful validation of the digital signature, using a mechanism as described in FPT_TUD_EXT.1.2.
FDEEEcPP20E:FPT_FUA_EXT.1.4
The TSF shall return an error code if any part of the firmware update process fails.

5.1.4.2 Protection of Key and Key Material (FDEAAcPP20E:FPT_KYP_EXT.1)

FDEAAcPP20E:FPT_KYP_EXT.1.1
The TSF shall [

- ***only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1(d) or encrypted, as specified in FCS_COP.1(g) or FCS_COP.1(e),***
- ***only store plaintext keys that meet any one of the following criteria [***
  - ***The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1]***

].

5.1.4.3 Protection of Key and Key Material (FDEEEcPP20E:FPT_KYP_EXT.1)

FDEEEcPP20E:FPT_KYP_EXT.1.1
The TSF shall [

- ***only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1(d) or encrypted, as specified in FCS_COP.1(g) or FCS_COP.1(e),***

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

].

5.1.4.4 Power Saving States (FDEAAcPP20E:FPT_PWR_EXT.1)

FDEAAcPP20E:FPT_PWR_EXT.1.1

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

5.1.4.5 Power Saving States (FDEEEcPP20E:FPT_PWR_EXT.1)

FDEEEcPP20E:FPT_PWR_EXT.1.1

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

(TD0464 applied)

5.1.4.6 Timing of Power Saving States (FDEAAcPP20E:FPT_PWR_EXT.2)

FDEAAcPP20E:FPT_PWR_EXT.2.1

For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur:

user-initiated request,

[***no other conditions***].

5.1.4.7 Timing of Power Saving States (FDEEEcPP20E:FPT_PWR_EXT.2)

FDEEEcPP20E:FPT_PWR_EXT.2.1

For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur:

user-initiated request,

[***no other conditions***].

5.1.4.8 TSF Testing (FDEAAcPP20E:FPT_TST_EXT.1)

FDEAAcPP20E:FPT_TST_EXT.1.1

The TSF shall run a suite of the following self- tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [***Cryptographic Algorithm Self-tests***].

5.1.4.9 TSF Testing (FDEEEcPP20E:FPT_TST_EXT.1)

FDEEEcPP20E:FPT_TST_EXT.1.1

The TSF shall run a suite of the following self- tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [***Cryptographic Algorithm Self-tests***].

5.1.4.10 Trusted Update (FDEAAcPP20E:FPT_TUD_EXT.1)

FDEAAcPP20E:FPT_TUD_EXT.1.1

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

FDEAAcPP20E:FPT_TUD_EXT.1.2

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

FDEAAcPP20E:FPT_TUD_EXT.1.3

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

5.1.4.11 Trusted Update (FDEEEcPP20E:FPT_TUD_EXT.1)

FDEEEcPP20E:FPT_TUD_EXT.1.1
Refinement: The TSF shall provide authorized users the ability to query the current version of the TOE [***firmware***].

FDEEEcPP20E:FPT_TUD_EXT.1.2
Refinement: The TSF shall provide authorized users the ability to initiate updates to TOE [***firmware***].

FDEEEcPP20E:FPT_TUD_EXT.1.3
Refinement: The TSF shall verify updates to the TOE [***firmware***] using a [***authenticated firmware update mechanism as described in FPT_FUA_EXT.1***] by the manufacturer prior to installing those updates.

5.2 TOE Security Assurance Requirements

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

Requirement Class	Requirement Component
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 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, list of all 3rd party software libraries (including version numbers), 3rd party hardware components (including model/version numbers)**].

ASE_TSS.1(c).1: Section **Error! Reference source not found.** 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 sowftware libraries and hardware components.

Software Library	Version Number	Description
at91-bootstrap	3.9.3	Bootloader
linux	5.4	Kernel
uClibc-ng	1.0.32	Libc (micro version of standard C library)
busybox	1.31.1	Userspace command line utilities
cryptsetup	2.2	Encryption of user configuration
evtest	1.34	Script access to discrete inputs
libnetfilter_conntrack	1.0.7	Firewall
netsnmp	5.8	SNMP Server
nftables	0.9.3	Firewall
openssh	8.4p1	SSH Server
openssl	1.1.1g	Cryptographic and SSL library

Table 3 3rd Party software Libraries/Packages

HW Component	Version/Part Number	Vendor
Microprocessor (CPU) system-in-package	ATSAMA5D27C-LD2G	Microchip
Integrated DRAM (CPU) system-in package	AD220032D	Microchip
eMMC Flash storage	SDINBDG4-8G	SanDisk
Ethernet adapter	KSZ8081RNA	Microchip
Power management	ACT8865	Qorvo
SATA HW encryption	X-Wall MX+-256C version 1.0	Enova
PCIe SATA controller	88SE9170	Marvell
Temperature sensor	TMP100MDBVREP	Texas Instruments

Table 4 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:

- FDEAAcPP20E:FCS_AFA_EXT.1: The TOE supports a password authorization factor, and the password may be up to 256 characters (bytes) in length and can be composed of uppercase letters, lowercase letters, numbers, and the "printable" special characters (found in section 5.1.1.28).
- FDEAAcPP20E:FCS_AFA_EXT.2: The TOE does not have any power-saving states beyond power-off. After transitioning from the power-off state, the user must authenticate before the TOE will allow data to be read from or written to the drive.
- FDEAAcPP20E:FCS_CKM.1(c): The TOE accept injection of externally supplied Key Database (KDB) which contains 256-bit DEKs encrypted with a KEK derived from user passphrases. The TOE allows the administrator to load the KDB while in provisioning mode.
- FDEAAcPP20E/FDEEEcPP20E:FCS_CKM.4(a): When the TOE powers off (as the TOE has no other power states other than off), all values in the integrated DRAM memory drain to a zero state. Temporary values are stored in memory and the interface used for key erasure is zeroization or overwrite.
- FDEEEcPP20E:FCS_CKM.4(b), FDEAAcPP20E:FCS_CKM.4(d): The TOE includes working memory RAM as part of its microprocessor, and this serves as the working memory in which the TOE uses the AES-KW key (derived from the password and salt) to unwrap working copies of loaded DEKs (unwrapped during authentication). The TOE clears all of these values from memory by overwriting them with zeros, after loading the DEKs into the hardware encryption engines. The TOE clears its hardware encryption engine registers either when explicitly commanded, or more typically upon a reset. The TOE also includes a discrete NAND Flash storage chip for persistent storage. The TOE interfaces with the Flash storage through its eMMC interface, performing all storage reads and writes through that interface. For keys stored (encrypted) within its Flash, to ensure that the sectors containing the encrypted KDB are erased, the TOE issues a Secure Erase eMMC command to the Flash, specifying the entire user configuration partition as the bounds of the Secure Erase.
- FDEEEcPP20E:FCS_CKM.4(e): The TOE performs a direct clearing (using the eMMC's Secure Erase command) of the user data partition, which contains the Key DataBase (i.e., the KDB) and the encrypted DEKs within.
- FDEAAcPP20E/FDEAAcPP20E:FCS_CKM_EXT.4(a): The TOE destroys plaintext key in volatile memory when no longer needed using zeroization or overwrite
- FDEAAcPP20E/FDEAAcPP20E:FCS_CKM_EXT.4(b): The TOE has the Compliant power saving state of G3 (Mechanical Off).
- FDEEEcPP20:FCS_CKM_EXT.6: The TOE clears its keys in accordance with FCS_CKM.4(b).
- FCS_COP.1: The TOE's Galleon Embedded Computing Encryption Module Firmware (version 1.00) contains cryptographic algorithm implementations that following NIST standards and has received the following CAVP algorithm certificates. Note that the TOE includes the Enova X-Wall AES-XTS hardware chip.

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(a) (Verify)	ECDSA P-384 w/SHA-384 Verify	FIPS 186-4, ECDSA	A2478
FCS_COP.1(b) (Hash)	SHA-384 Hashing	FIPS 180-4	A2478
FCS_COP.1(c) (Keyed Hash)	HMAC-SHA-384	FIPS 198-1 & 180-4	A2478
FCS_COP.1(d) (Key Wrap)	AES-256 KW	FIPS 197, SP 800-38F	A2478
FCS_COP.1(f) (AES)	AES-256 XTS Encrypt/Decrypt	FIPS 197	AES 4013
FCS_COP.1(f) (AES) FCS_VAL_EXT.1(Validation)	AES-256 CBC Encrypt/Decrypt	FIPS 197	A2478
FCS_RBG_EXT.1 (Random)	AES-256 CTR_DRBG	SP 800-90A	A2478

Table 5 Cryptographic Algorithms

- FDEAAcPP20E/FDEEEcPP20E:FCS_COP.1(a): The TOE utilizes ECDSA P-384 with SHA-384 signatures to verify the authenticity of firmware updates. Upon receiving a candidate update and the accompanying signature file, the TOE uses an embedded vendor public key to verify the ECDSA signature against the received image. The verification uses SHA-384 and follows the FIPS 186-4 ECDSA format.
- FDEAAcPP20E/FDEEEcPP20E:FCS_COP.1(b): The TOE implements the SHA-384 algorithm and uses the SHA-384 algorithm as part of PBKDFv2 key derivation. The TOE uses SHA-384 hashing when verifying trusted update ECDSA P-384 signatures and then calculating HMAC-SHA-384 checksums.
- FDEAAcPP20E/FDEEEcPP20E:FCS_COP.1(c): The TOE implements HMAC-SHA-384 using a 384-bit key, the SHA-384 hash algorithm, a 1024-bit block size, and an output MAC length of 384-bits. The TOE uses its HMAC-SHA-384 implementation during PBKDFv2 key derivation.
- FDEAAcPP20E/FDEEEcPP20E:FCS_COP.1(d): The TOE uses AES-KW (compliant with NIST SP 800-38F) to decrypt (unwrap) the DEKs stored in its Key DataBase (KDB) stored within its Flash memory.
- FDEAAcPP20E/FDEEEcPP20E:FCS_COP.1(f): The TOE possesses AES XTS implementations dedicated to drive encryption/decryption. The TOE's implementation exclusively uses 256-bit keys. The TOE also utilizes an AES CBC implementation when verifying a user's password (by deriving a PBKDFv2 key from the user's password, and then using that key to decrypt a user tag, ensuring that the resulting plaintext matches the expected, fixed string).
- FDEAAcPP20E/FDEEEcPP20E:FCS_KDF_EXT.1: The TOE uses an 800-132 PBKDF in counter mode using SHA-256 or SHA-384 and 10,000 iterations and a 256-bit salt to transform the operator's password into a key for unwrapping.
- FDEAAcPP20E/FDEEEcPP20E:FCS_KYC_EXT.1/2: The TOE uses PBKDFv2 to transform the operator's password into a 256-bit BEV and then uses that BEV as described in FCS_VAL_EXT.1 below.
- FDEAAcPP20E:FCS_PCC_EXT.1: TOE allows user and an administrative passwords up to 256-bytes (characters in length), uppercase/lowercase letters, numbers, and the following characters *" ", "!", "", "#", "$", "%", "&", "'", "(", ")", "*", "+", ",", "-", ":", ";", "<", "=", "@", "[", "\", "]", "^", "{", "|", "}"*. The TOE will reject a password containing other characters. The TOE conditions passwords by combining them with a 256-bit salt using PBKDFv2.
- FDEAAcPP20E/FDEEEcPP20E:FCS_RBG_EXT.1: The TOE includes an AES-256 CTR_DRBG that it seeds with at least 256-bits of entropy from two software-based noise sources.
- FDEAAcPP20E/FDEEEcPP20E:FCS_SNI_EXT.1: The TOE generates its salts (the admin account has a 256-bit salt used during PBKDFv2 derivation) and AES-CBC IVs using its AES-256 CTR_DRBG when generating a user's random 256-bit salt and then XORing that value in half to create the AES CBC decryption IV. While the TOE does not generate any nonces, it does generate AES XTS tweak values using its CTR_DRBG.
- FDEEEcPP20E:FCS_VAL_EXT.1: A user password is required when the machine is power cycled. The TOE validates the operator's password by verifying that the PBKDFv2 key derived from the user's password can successfully AES CBC decrypt the encrypted value of a known string (the TOE creates the IV by XORing the user's random 256-bit salt together into a 128-bit value). If the verification operation fails, then the TOE treats this as an invalid login and increments its failed login attempts counter for that user.

If the counter reaches five, the TOE resets itself, which results in an approximate twenty-second delay, until the EM has fully booted, and will accept another authentication attempt. 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:

- FDEEEcPP20E:FDP_DSK_EXT.1: The TOE provides hardware-based FDE and encrypts the entirety of its attached drives through an AES-256 XTS block based encryption. The TOE sits on the SATA path between the NAS and its attached drives. Because of its position, the TOE guarantees that all data written to and read from the data drives are encrypted. The Admin Guide describes the TOE's initialization process and setup for the HW-layer. The TOE maintains separate, unencrypted, internal Flash memory partitions to house its firmware. This lies beyond the RMC drives 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:

- FDEAAcPP20E:FMT_MOF.1: The TOE claims no Compliant power saving states beyond power off and only an authorized user can command the TOE to power off.
- FDEAAcPP20E/FDEEEcPP20E:FMT_SMF.1: The TOE allows an administrator to change a DEK (by importing a new Key DataBase [KDB], which overwrites/changes all DEKs, while provisioning or re-provisioning the TOE), import wrapped DEKs (again, by importing a new KDB), and initiate a TOE firmware update. The TOE supports changing of the authorization factor (the administrator can change his or her own password, and may change the password of a user by loading a new KDB, created using the user's new password).
- FDEAAcPP20E: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:

- FDEEEcPP20E :FPT_FUA_EXT.1: The TOE uses an internal ECDSA P-384 public key (hardcoded within the TOE's existing firmware image, stored within the microprocessor) to verify new firmware images before writing the firmware to the TOE's internal storage. The TOE maintains both a primary and secondary partition for images, as well as a third, recovery partition (in the event that both the primary and second partitions become corrupted).
- FDEAAcPP20E/FDEEEcPP20E:FPT_KYP_EXT.1: The TOE stores keys persistently in the KDB stored within its user data partition residing on its internal Flash memory. The KDB contains all keys in encrypted form (encrypted with an AES-KW key derived from a user's password plus an internal salt). The TOE also stores an AES XTS plaintext key; however, the TOE does not use this key to encrypt any user data, but rather uses it to decrypt its own TOE data and configuration, which the TOE stores in internal Flash.
- FDEAAcPP20E/FDEEEcPP20E: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 computer/server containing the EM. The TOE must be fully rebooted from this state.
- FDEAAcPP20E/FDEEEcPP20E:FPT_TST_EXT.1: The TOE includes the following power-up Known Answer Tests (KATs) to ensure that each of its cryptographic algorithms operates correctly.
  - ECDSA verify test
  - SHA hashing tests
  - HMAC-SHA hashing test

- AES CBC encryption/decryption test
- AES XTS encryption/decryption test
- AES-256 CTR_DRBG tests (including SP 800-90A section 11.3 health tests)

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

- FDEAAcPP20E/FDEEEcPP20E:FPT_TUD_EXT.1: The TOE allows updates to the HW-layer's firmware while in provisioning mode. The TOE only accepts updates that have been signed and delivered by Galleon Embedded Computing. The TOE protects and maintains its update credential (the update public key) by storing it within its read-only firmware image. The TOE provides no access to this public key, and only a Galleon signed update (were to include a new public key) can replace the key. The TOE will verify the ECDSA P-384 with SHA-384 signature of the update image to ensure authenticity, and if valid, the TOE will update its firmware. The TOE will automatically validate the update and not continue if the embedded digital signature in the update package cannot be verified.