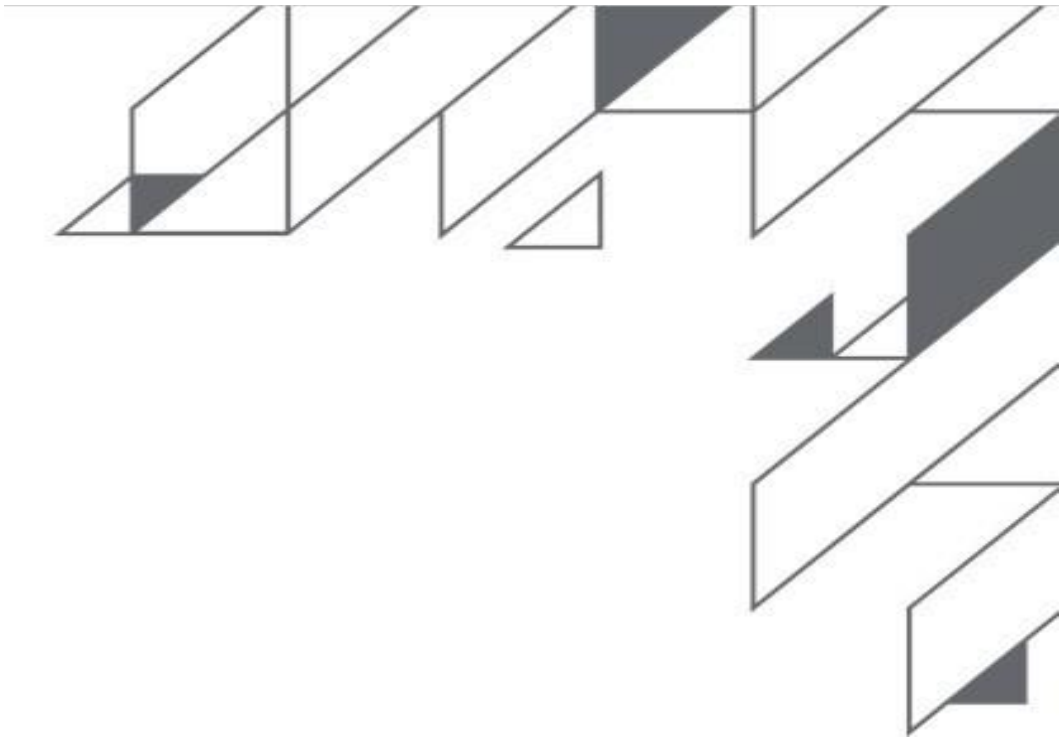

MARVELL

Marvell LS2 HSM Security Target

Date: 20/02/2026

Change History

Version	Date	Author	Comment
1.0	08/05/2024	Marvell Semiconductor, Inc.	First draft
1.1	8/10/2024	Marvell Semiconductor, Inc.	Second draft
1.2	16/10/2024	Marvell Semiconductor, Inc.	Third draft
1.3	25/10/2024	Marvell Semiconductor, Inc.	Fourth draft
1.4	06/11/2024	Marvell Semiconductor, Inc.	Fifth draft
1.5	11/01/2025	Marvell Semiconductor, Inc.	Sixth draft.
1.6	10/03/2025	Marvell Semiconductor, Inc.	Seventh draft
1.7	13/03/2025	Marvell Semiconductor Inc.	Eighth draft
1.8	27/03/2025	Marvell Semiconductor Inc.	Ninth draft
1.9	24/04/2025	Marvell Semiconductor Inc.	Tenth draft
1.10	01/05/2025	Marvell Semiconductor Inc.	Eleventh draft
1.11	17/06/2025	Marvell Semiconductor Inc.	Twelfth draft
1.12	03/07/2025	Marvell Semiconductor Inc.	Thirteenth draft
1.13	22/08/2025	Marvell Semiconductor Inc.	Fourteenth draft
1.14	21/11/2025	Marvell Semiconductor Inc.	Fifteenth draft
1.15	05/12/2025	Marvell Semiconductor Inc.	Sixteenth draft
1.16	08/12/2025	Marvell Semiconductor Inc.	Seventeenth draft
1.17	14/01/2026	Marvell Semiconductor Inc.	Eighteenth draft
1.18	20/02/2026	Marvell Semiconductor Inc.	First release

List of Tables

1 ST INTRODUCTION

1.1 ST REFERENCE

Title: Marvell LS2 HSM Security Target

Version: v1.18

Author: Marvell Semiconductor, Inc.

Date of publication: 20/02/2026

1.2 TOE REFERENCE

TOE Name: Marvell LS2 HSM

TOE Developer: Marvell Semiconductor, Inc.

TOE version:

Hardware version	Firmware version	Bootloader version
CN9310410-03 C	MARVELL-LS2-FW-10.24-0780	MARVELL-LS2-UBOOT-10.24-0702-R01-SB* MARVELL-LS2-UBOOT-10.24-0702-R02-SB*

* R01 and R02 both use same bootloader code however the private key used to sign the bootloader image is different so the different naming.

1.3 TOE OVERVIEW

INTRODUCTION

The LS2 Hardware Security Module (hereafter referred to as TOE) by Marvell is a high-performance purpose-built security solution for key management and crypto acceleration. The module is deployed in a PCIe slot to provide crypto and protocol acceleration in a secure manner to the system host. It is typically deployed in a server or an appliance to provide crypto offload for the keys stored on the HSM. The module's functions are accessed over the PCIe interface via an API defined by the module.

The HSM is a hardware multi-chip embedded cryptographic module with firmware programmed on the HSM. The module provides cryptographic primitives to accelerate SOGIS approved (refer to **[SOG-IS-Crypto]**) algorithms to support use cases including PKI, Code Signing, Document Signing, Root of Trust, and TLS. The cryptographic functionality includes asymmetric (RSA/EC) and symmetric (AES) ciphers, signatures, and random number generation, along with protocol-specific complex instructions to support TLS 1.2. The module implements password-based single-factor authentication. The physical boundary of the module is the outer perimeter of the card itself.

TOE TYPE

The TOE is a Hardware Security Module (HSM) in the form of a PCI-e card, composed of hardware, firmware, and software.

The TOE is suitable for use by trust service providers supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services (including support of authentication of client applications or authorized users of secret keys, and support of authentication for electronic identification), as identified by the (EU) No 910/2014 regulation of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (eIDAS) in [Regulation]. The TOE may also support protected backup of keys.

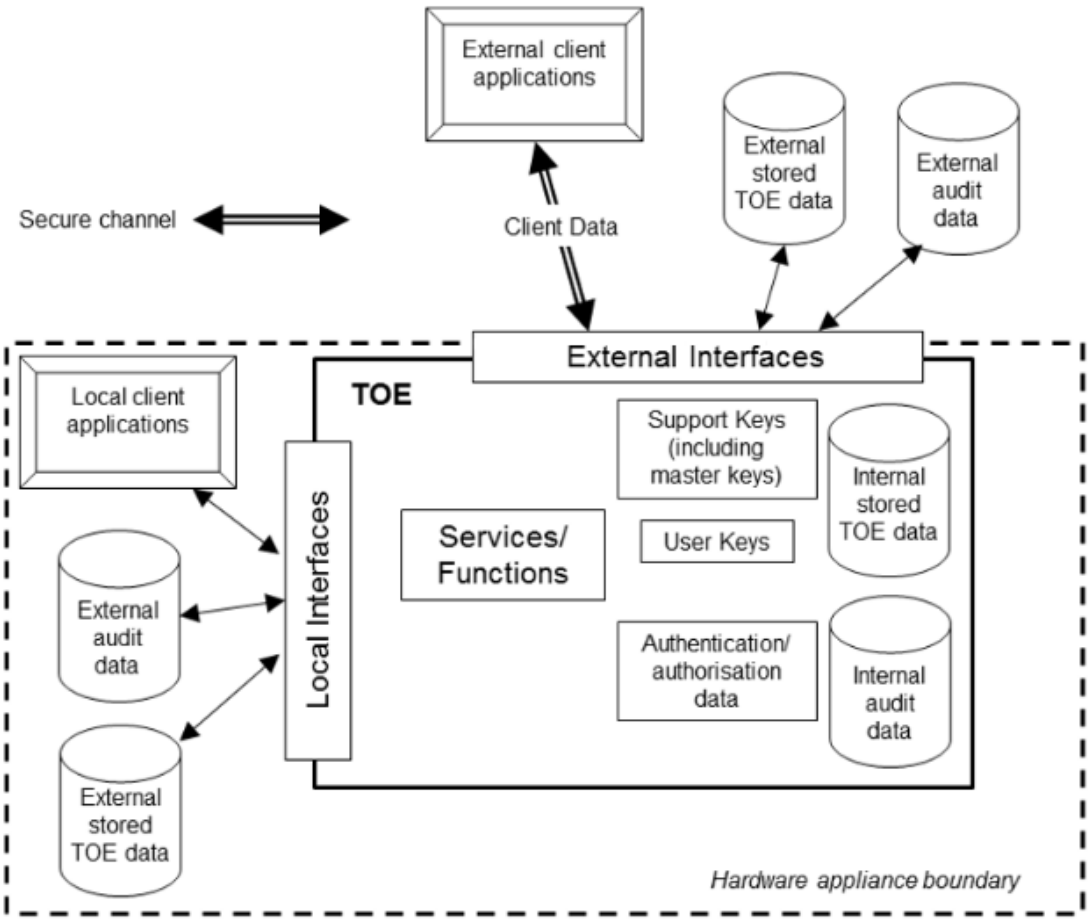

Figure 1 Generic TOE Architecture

TOE USAGE & MAJOR SECURITY FEATURES

The TOE provides a physically and logically protected component for the performance of cryptographic functions, providing the following major security features:

- Cryptographic functions including symmetric and asymmetric algorithms approved by [SOG-IS-Crypto] and for use in TSP environments for encryption, decryption, digital signature, verification and hash generation.

- Key management intended as key generation, key import and export and key derivation.
- Backup of the critical user data.
- Audit of the security relevant events of the TOE, to be stored within the TOE and securely exported to a third-party application located on the host.
- Self-protection, which includes self-tests and physical protection.
- Secure communication channels for remote management operations.
- Information and functionality isolation provided by HSM partitions.

The TOE comprises two processors, a local system controller, ROM, FLASH memory, DRAM, and a microcontroller with backup batteries embedded in a tamper resistant environment.

It is connected to a host system through a Gen4x8 PCIe interface, which is used to access TOE services by the use of the dedicated APIs and SDK.

When installing the TOE, the SO must then set the configurable policies at the cryptographic module level and create at least one partition, with its corresponding user in the Crypto Officer role to make the cryptographic module ready for use.

The security operator may also be required to make policy settings at the partition level to conform to the organization's security requirements. In addition, the partitions should be initialized for CC compliant operation. In operation, the TOE requires users in any of the three (3) roles to be identified and authenticated before they are authorized to perform any cryptographic and/or key management operations.

Authentication is performed using a one-time challenge-response mechanism for the Crypto Officer and Crypto User roles.

NON-TOE HARDWARE/SOFTWARE/FIRMWARE

The TOE is a Cryptographic Module which comprises its own hardware and software.

It is installed in a larger Non-TOE system (typically a GPC or a server) as host device, that shall meet the following requirements:

Hardware	Requirement
System architecture	x86 (AMD or Intel)
PCIe	Low-profile (HHHL) PCIe Gen4x8
Virtualization support	SR-IOV support enabled.

Table 1 Hardware Requirements

Operating System	Requirement
Linux	CentOS 8.3

	Ubuntu 18.04-5-LTS (Bionic Beaver)
	Ubuntu 20.04

Table 2 Supported Operating Systems

1.4 TOE DESCRIPTION

INTRODUCTION

The TOE is a separate component with its own hardware and software, communicating via physical and logical interfaces with the client application remotely or in the TSP system. The physical interfaces used to connect the TOE to client applications are the PCIe VFs.

The threat environment the TOE is designed for is one of high threat of network compromise, and low threat of physical compromise (for example, a Certification Authority facility with a high degree of physical protection, but an operational requirement to be connected to an untrusted network such as the internet).

The environment is assumed to prevent prolonged unauthorized physical access to the TOE (including theft). The TOE provides physical protection mechanisms to deter undetected compromise of its security functions by low attack potential individuals that do have physical access to the TOE (for example disgruntled employees with legitimate access to the TOE).

The TOE is responsible for protecting the keys against logical attacks that would result in disclosure, compromise and unauthorized modification, and for ensuring that the TOE services are only used in an authorized way.

Client applications request cryptographic functions from the TOE once the appropriate authorization has been provided.

The TOE performs local cryptographic operations, and associated key management, which can be used by an application using server signing to create qualified electronic signatures and qualified electronic seals on behalf of a legal or natural person which is distinct from and remote from the TSP which manages the TOE. The TOE generates, stores and uses signing/sealing keys in a way that maintains the remote control of an identified signatory or seal creator who operates through the use of a client application. The TOE deals with ensuring the security of keys and their use for signature or seal creation.

The TOE logical boundary is defined by the logical components which support the security functionality:

- Bootloader
- Firmware

TOE LOGICAL SCOPE

The security functionalities of the TOE are:

Cryptographic functions

The TOE include cryptographic algorithms to provide the following cryptographic functions:

- Digital signature generation and verification
- Message digest generation
- Message authentication code generation and verification
- Encryption and decryption (symmetric and asymmetric)
- Key generation (symmetric and asymmetric)
- Key agreement and distribution
- Key derivation
- Generation of shared secret values
- Cryptographic support (ability of TOE to be used for doing any operation required as part of an OTP mechanism or PKI infrastructure, and Non-PKI is usually credential mechanisms) for one time password and other non-PKI based authentication mechanisms
- Random number generation.

The cryptographic algorithms used to perform these services belong to the approved algorithms in **[SOG-IS-Crypto]** and are used to support TSP system functions to create electronic seals or electronic timestamps apart from other non-TSP services.

Key management

The TOE supports the secure management of cryptographic keys for those implemented cryptographic functions which requires them, including:

- Key establishment (including key generation)
- Protection of keys held within the TOE and held externally (for use by the TOE);
- Control of access and use of keys by the cryptographic functions within the TOE
- Deletion of keys within the TOE.

The TOE supports the following techniques for establishing keys:

- Generation of cryptographic keys using a random number generator and implementing the key generation algorithms depending on the intended use of the keys.
- Import of cryptographic keys in encrypted form.
- Key agreement protocols establishing common secrets with external entities.
- Derivation of keys from shared knowledge.

Secret keys are associated with attributes that determine their use, such that the correct association between the key and its attributes are protected against unauthorized modification. The key attributes maintained by the TOE allow to determine the following characteristics:

- Identify the key (this enables it to be linked to a particular owner)

- The type of the key (e.g. whether the key is a secret key of a symmetric cryptographic algorithm or the secret (commonly called private) key of an asymmetric cryptographic algorithm)
- Owner of the key (required only for secret keys)
- Usage of the key (to determine which cryptographic functions that can use the key)
- Whether the key is allowed to be exported
- Whether the key is an Assigned Key
- Integrity protection data that protects the integrity of the key value, the values of the key attributes, and the binding of the key to its attributes.

In this TOE, authorization to change the attributes of a key is distinct from authorization to use the key for cryptographic functions. This is supported by the definition of an 'Assigned Key' which cannot be imported or exported, for which the re-authorization conditions and key usage cannot be changed, and for which the authorization data can only be changed on successful validation of the current authorization data.

Keys can leave the TOE in one of three possible situations:

- External storage of keys

The TOE allows external storage of keys for later use by the TOE (or another instance of the TOE within the same authorized security infrastructure operated by a TSP) as 'external stored TOE data', for example to deal with large numbers of keys when the TOE does not have sufficient internal storage to hold them all internally. The form in which the keys are stored is sufficient to protect the confidentiality and integrity of the keys and the binding of the keys to their attributes.

- Export of keys

The TOE allows export of keys for use by authorized client applications, provided that they are not Assigned Keys, that other key attributes do not prohibit export, and that the correct authorization data for the key has been supplied (through user authentication). Although the TOE checks key attributes to determine whether to allow export, the appropriate values to use for the key attributes will depend on the application context in which the key is used, and the security measures (technical, physical, and procedural) that apply to that context. For example, in the case of some future Post Quantum Signature algorithms export of the private key will not be permissible.

Keys might be imported or exported as part of providing general cryptographic functions (e.g., in support of client applications that use the TOE to support their own authentication mechanisms), but the TOE also allows individual secret keys to be identified as non-exportable. Assigned keys cannot be imported or exported and represent a more strongly controlled type of key that is intended to be used only within the TOE for operations such as electronic signature or electronic seal generation.

Applications can use the TOE for key pair generation and offload private key operation to the TOE and share the corresponding public key to third-party application by exporting public keys.

- Backup

The TOE provides facilities for secure backup and restores of the TSF state which includes user keys and internal support keys used to protect the TSF.

Backup

The TOE supports backup and restoration of the TSF state necessary to re-establish an operational state after failure to specially preserve the security requirements on keys.

The confidentiality and integrity of the data stored in the backup, especially the keys, is protected with a level of protection according to the one provided when being inside the TOE.

The availability of this backup is the responsibility of the operational environment, which also will operate under strong environmental and procedural controls. The corresponding 'restore' operation will only be carried out under at least dual person control.

Audit

The TOE also logs audit records for its own actions, which can be collected, maintained, and reviewed in a larger system from which the TOE is part.

The TOE includes the Appliance User (AU) role as the unique role allowed to access the logs and export them.

Self-protection

The module is designed with self-protection capabilities that ensure that it remains in a secure state when facing abnormal conditions. The following capabilities are implemented by the module in order to maintain a secure operational state:

- The module is coated in hard epoxy, such that any physical breach attempt leaves behind evidence of tamper.
- Self-testing capability that ensures firmware integrity and correct performance of cryptographic algorithms and random number generators.
- Protection against physical attacks through monitoring voltage and temperature.
- Protection against power loss.

Secure channels

Remote operation of the TOE is only allowed for TOE Partitions. The end-to-end encryption feature in the module allows an application to initiate a TLSv1.2 connection with the firmware to ensure the confidentiality of the data communicated to the HSM. It has also a provision to have a Hybrid Key exchange schemes are based on combining Elliptic Curve Diffie-Hellman (ECDH) with a post-quantum key encapsulation method (PQ KEM) using the existing TLS PRF.

Authentication & authorization

The module enforces identity-based authentication. A role is explicitly selected at authentication; the MCO role is associated with the Master Partition and the PCO and PCU roles are associated with user partitions. The module allows one identity per role, per partition.

Two types of authorization are used by the TOE:

- Role-based authorization allows users to access a given set of functionalities based on their role.
- Keys authorization allows users to make use of the keys they are authorized to use. (e.g., only the owner of a key shall be authorized to sign using the given key).

Partitions

Cryptographic keys are stored and managed inside containers called partitions. The TOE supports two types of partitions:

- Master partition: It is a single, mandatory partition whose main purpose is to support administrative tasks at the HSM level. Specific roles are defined for the partition, and there can be only one Master partition inside the TOE.
- User partitions: optional partition whose main purpose is to group all keys belonging to a same entity or application in a dedicated container isolated from the other partitions. Specific roles are supported at the user partition level, which are different from the roles defined in the Master partition.

Any type of partition (Admin partition or User partition) can contain cryptographic keys. For a given partition, the management and usage of the related key material is restricted to the roles assigned to that partition, therefore enforcing strict isolation between the different partitions managed inside the TOE.

The LS2 HSM adapter is an SR-IOV enabled intelligent PCIe adapter with one physical function and 64 virtual functions (VFs). In addition to crypto offloads, this adapter can provide secure key storage with up to 42 logical partitions, including the masterpartition. Each partition has dedicated resources that are logically and securely isolated from other partitions. A partition has its own specific policies and controls, and its own user accounts to manage what can be done with a partition and by a user of a partition. Consequently, each partition is treated as a virtual HSM and referred to as a pHSM (or HSM Partition). The TOE always has one default partition called the Master Partition, which exposes the interfaces to create, delete, resize, clone and update the remaining Partitions.

The TOE provides isolated internal paths between the host and the addressed partition to manage each partition request.

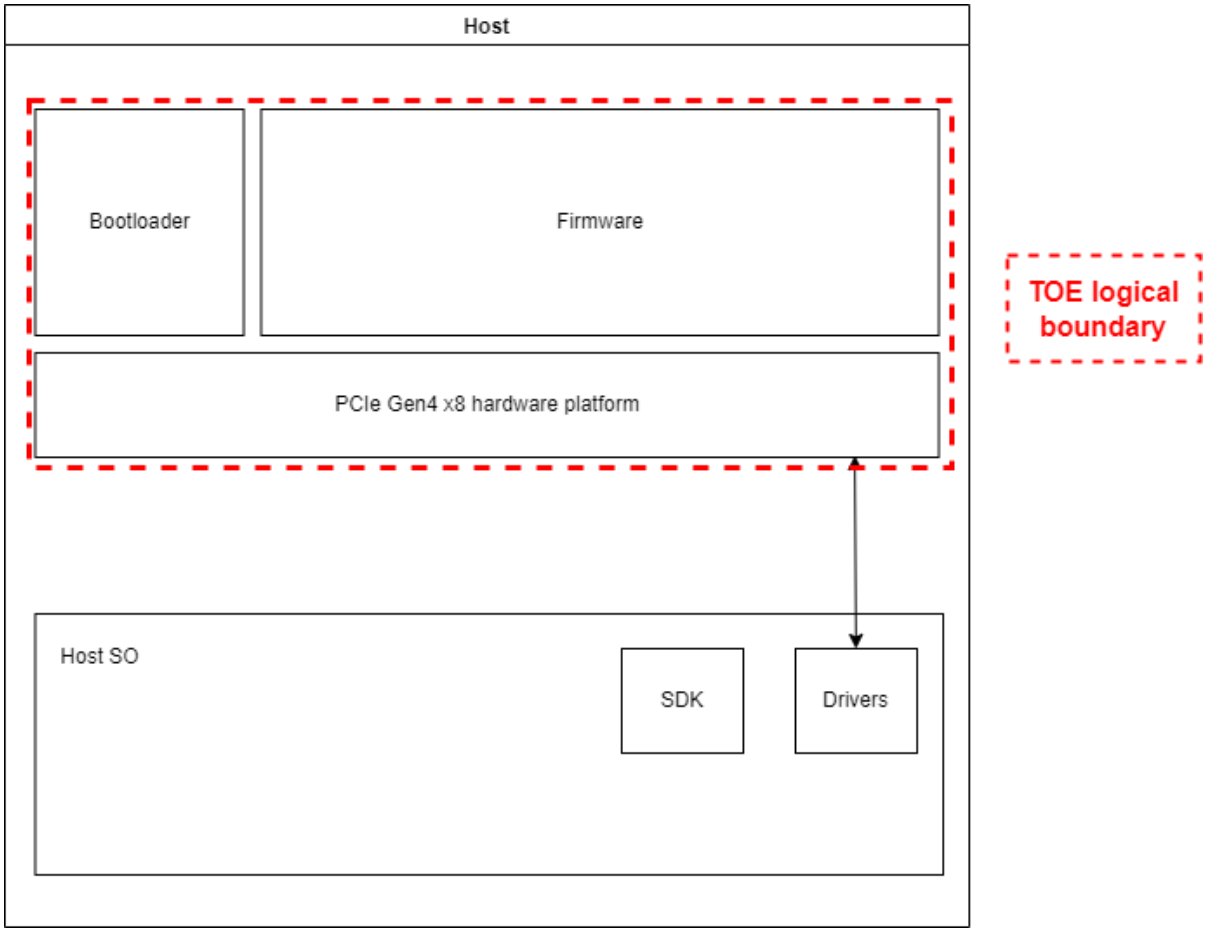

Figure 2 TOE logical boundary

Guidance's user to bring the TOE to operational state and to operate the TOE in a secure manner, are not considered as part of the logical boundary but the physical.

TOE PHYSICAL SCOPE

The physical boundary of the TOE is an electronic board in the shape of a PC expansion card ready to be plug on a X8 lane Gen4 host side PCIe.

The hardware can be identified with the part version of CN9310410-03 C.

The PC card contains:

- The board embodiments with the circuitry partially covered by potting material and heatsink.
- The epoxy enclosure of the module prevents physical access to any of the internal components without having to destroy the module.
- Hardware interfaces for interact with the board, as they are:

- X8 lane Gen4 PCIe interface to provide power and communication interface with host device.
- 10 GbT interface (not in use)
- Zeroize button as a direct zeroization interface
- LED 1 status LED used to inform the user about the results of the zeroization
- LED 2 status LED used to convey the state of the adapter
- UCD LED to indicate the physical condition of the device (operational or fault).

Figure 3 TOE physical boundary

Physical components and delivery

The hardware part of the TOE is delivered to the client in a labelled box with a trusted courier. The software components and guidance's can be downloaded through Marvell's file transfer platform, which access is defined for the client's profile. The board is embedded with both Firmware and bootloader (in different images). If the firmware and bootloader with which the board is distributed are not the evaluated ones, they are updated by the tested version through a trusted electronic delivery.

Item	Name	Version	Type	Delivery method
Board	LS2-B0	CN9310410-03 C	Security bag in a labelled and opaque package	Trusted courier
Software	Firmware	MARVELL-LS2-FW-10.24-0780	Tar file (.tgz)	Electronic delivery
	Bootloader	MARVELL-LS2-UBOOT-10.24-0702-R01-SB MARVELL-LS2-UBOOT-10.24-0702-R02-SB	Tar file (.tgz)	Electronic delivery
Document	Marvell LS2 HSM Hardware Installation	Rev.3	pdf file	Electronic delivery

	Guide			
	Marvell LS2 HSM Adapter SDK User Guide	Release 10.24-0780 Rev.1.1	pdf file	Electronic delivery
	Marvell LiquidSecurity 2 SDK API Guide	10.24-0780	ZIP file(.zip)	Electronic delivery
	Marvell LS2 HSM User Guidance	Version 1.6	Pdf file	Electronic delivery
	Marvell LS2 HSM Preparative Procedures	Version 6.0	pdf file	Electronic delivery

Table 3 TOE physical boundary components

2 CONFORMANCE CLAIMS

Common Criteria version: This ST conforms to

“Common Criteria for Information Technology Security Evaluation”, CC:2022 Release1, Parts 1-5 ([CC2022P1R1], [CC2022P2R1], [CC2022P3R1], [CC2022P4R1], [CC2022P5R1])

“Common Methodology for Information Technology Security Evaluation: Evaluation Methodology”, CEM 2022 Release1 [CEM2022R1].

Conformance for this ST is claimed for Common Criteria Part 2 conformant and Common Criteria Part 3 conformant.

The assurance requirement of this Security Target is EAL4 + AVA_VAN.5, ALC_FLR.3

Protection Profile (PP) conformance claim:

This ST claims strict conformance to the **[CEN EN 419221-5]** protection profile.

This ST includes additional requirements in order to claim isolation of partitions**.**

Isolation Requirements:

- FDP_IFC.1/Partitions
- FDP_IFF.1/Partitions

3 SECURITY PROBLEM DEFINITION

This section describes the security aspects of the operational environment and its expected use in said environment. It includes the declaration of the TOE operational environment that identifies and describes:

- The alleged known threats that will be countered by the TOE
- The organizational security policies that the TOE has to adhere to
- The TOE usage assumptions in the suggested operational environment.

We will begin defining Assets and Agents of threats.

3.1 ASSETS

R.SecretKey: secret keys used in symmetric cryptographic functions and private keys used in asymmetric cryptographic functions, managed and used by the TOE in support of the cryptographic services that it offers. This includes user keys, owned and used by specific users, and support keys used in the implementation and operation of the TOE. The asset also includes copies of such keys made for external storage and/or backup purposes. The confidentiality and integrity of these keys must be protected.

R.PubKey: public keys managed and used by the TOE in support of the cryptographic services that it offers (including user keys and support keys). This asset includes copies of keys made for external storage and/or backup purposes. The integrity of these keys must be protected.

R.ClientData: data supplied by a client for use in a cryptographic function. Depending on the context, this data may require confidentiality and/or integrity protection.

R.RAD: reference data held by the TOE that is used to authenticate an administrator (hence to control access to privileged administrator functions such as TOE backup, export of audit data) or to authorize a user for access to secret and private keys (R.SecretKey). This asset includes copies of authentication/authorisation data made for external storage and/or backup purposes. The integrity of the RAD must be protected; its confidentiality must also be protected unless the authentication method used means that the RAD is public data (such as a public key).

AS.P_MEM: parts of the firmware whose execution is reserved to the master partition.

AS.P_CPT_ENGINE : CPT cores protected in confidentiality and integrity when performing cryptographic operations.

AS.P_RAM: Portions of RAM assigned to specific partitions protected in confidentiality and integrity.

3.2 THREAT AGENTS

ATTACKER: a subject who is not authorised for the relevant action, but who may present themselves as either a completely unknown user, or as one of the subjects identifies in this ST (but in this case the attacker will not have access to the authentication or authorisation data for the subject).

PARTITION_USER: a user of an existing User Partition that attempts against assets from a different partition than its own.

3.3 THREATS TO SECURITY

This section identifies the threats to assets that require protection by the TOE. The threats are defined in terms of assets concerned, attackers and the adverse action that materializes the threat.

T.KeyDisclose **Unauthorised disclosure of secret/private key**

An attacker obtains unauthorised access to the plaintext form of a secret key (R.SecretKey), enabling either direct reading of the key or other copying into a form that can be used by the attacker as though the key were their own. This access may be gained during generation, storage, import/export, use of the key, or backup if supported by the TOE.

T.KeyDerive **Derivation of secret/private key**

An attacker derives a secret key (R.SecretKey) from publicly known data, such as the corresponding public key or results of cryptographic functions using the key or any other data that is generally available outside the TOE.

T.KeyMod **Unauthorised modification of a key**

An attacker makes an unauthorised modification to a secret or public key (R.SecretKey or R.PubKey) while it is stored in, or under the control of, the TOE, including export and backups if supported. This includes replacement of a key as well as making changes to the value of a key, or changing its attributes such as required authorisation, usage constraints or identifier (changing the identifier to the identifier used for another key would allow unauthorised substitution of the original key with a key known to the attacker). The threat therefore includes the case where an attacker is able to break the binding between a key and its critical attributes.

T.KeyMisuse **Misuse of a key**

An attacker uses the TOE to make unauthorised use of a secret key (R.SecretKey) that is managed by the TOE (including the unauthorised use of a secret key for a cryptographic function that is not permitted for that key), without necessarily obtaining access to the value of the key.

T.KeyOveruse **Overuse of a key**

An attacker uses a key (**R.SecretKey**) that has been authorised for a specific use (e.g. to make a single signature) in other cryptographic functions that have not been authorised.

T.DataDisclose **Disclosure of sensitive client application data**

An attacker gains access to data that requires protection of confidentiality (**R.ClientData**, and possibly **R.RAD**) supplied by a client application during transmission to or from the TOE or during transmission between physically separate parts of the TOE.

T.DataMod **Unauthorised modification of client application data**

An attacker modifies data (**R.ClientData** such as DTBS/R, authentication/authorisation data, or a public key (**R.PubKey**)) supplied by a client application during transmission to the TOE or during transmission between physically separate parts of the TOE, so that the result returned by the TOE (such as a signature or public key certificate) does not match the data intended by the originator of the request.

T.Malfunction **Malfunction of TOE hardware or software**

The TOE may develop a fault that causes some other security property to be weakened or to fail. This may affect any of the assets and could result in any of the other threats being realised. Particular causes of faults to be considered are:

- environmental conditions (including temperature and power)
- failures of critical TOE hardware components (including the RNG)
- corruption of TOE software.

T.P_CROSS_ACCESS **Partition cross access to resources**

The TOE offers user partitions. All users/assets (partition keys)/resources belonging to one user partition shouldn't be accessed by another user partition.

As this TOE has multiple partitions, unauthorized access to the partition assets is a threat.A **PARTITION_USER** access an asset (**AS.P_MEM**, **AS.P_CPT_PROCESS** , **AS.P_RAM**) belonging to a different partition to which the **PARTITION_USER** is not allowed to access.

T.P_EXECUTION **Master partition code execution**

The TOE offers a master partition which does the administrative work for the TOE. This partition should have dedicated resources which other user partitions shouldn't have access.

As this TOE has Master partition, this threat must be addressed.

A **PARTITION_USER** executing parts of the code reserved for the Master Partition (**AS.P_MEM**).

3.4 ORGANIZATIONAL SECURITY POLICIES

The organizational Security policies are defined as follows.

P.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized authorities as appropriate for use by TSPs.

P.KeyControl Support for control of keys

The life cycle of the TOE and any secret keys that it manages (where such keys are associated with specific entities, such as the signature creation data associated with a signatory or the seal creation data associated with a seal creator), shall be implemented in such a way that the secret keys can be reliably protected by the legitimate owner against use by others, and in such a way that the use of the secret keys by the TOE can be confined to a set of authorized cryptographic functions.

P.RNG Random Number Generation

The TOE is required to generate random numbers that meet a specified quality metric, for use by client applications. These random numbers shall be suitable for use as keys, authentication/authorization data, or seed data for another random number generator that is used for these purposes.

P.Audit Audit trail generation

The TOE is required to generate an audit trail of security-relevant events, recording the event details and the subject associated with the event.

3.5 ASSUMPTIONS

The assumptions when using the TOE are the following:

A.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities must provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment. In particular, any backups of the TOE and its data are maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data does not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data requires at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

A.Env Protected operating environment

The TOE operates in a protected environment that limits physical access to the TOE to authorised Administrators. The TOE software and hardware environment (including client applications) is installed maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment.

A.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE will ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application will ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and will correctly and securely manage the signature received from the TOE; and when certifying a public key the client application will ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) performs a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS. Client applications are also responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events. Similar requirements apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys. Appropriate procedures are defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

A.Uauth Authentication of application users

Any client application using the cryptographic services of the TOE will correctly and securely gather identification and authentication/authorisation data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorisation data as required) when required to authorise the use of TOE assets and services.

A.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

A.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data will be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

4 SECURITY OBJECTIVES

The security objectives are high level declarations, concise and abstract of the solution to the problem exposed in the former section, which counteracts the threats and fulfills the security policies and the assumptions.

These consist of:

- the security objectives for the operational environment.
- the security objectives for the TOE

4.1 SECURITY OBJECTIVES FOR THE TOE

The security objectives for the TOE must determine (to the desired extent) the responsibility of the TOE in countering the threats and in enforcing the OSPs. Each objective must be traced back to aspects of identified threats to be countered by the TOE and to aspects of OSPs to be met by the TOE.

OT.PlainKeyConf Protection of confidentiality of plaintext secret keys

The plaintext value of secret keys is not made available outside the TOE (except where the key has been exported securely in the manner of OT.ImportExport). This includes protection of the keys during generation, storage (including external storage), and use in cryptographic functions, and means that even authorized users of the keys and administrators of the TOE cannot directly access the plaintext value of a secret key.

OT.Algorithms Use of approved cryptographic algorithms

The TOE offers key generation functions and other cryptographic functions provided for users that are endorsed by recognized authorities as appropriate for use by TSPs. This ensures that the algorithms used do not enable publicly known data to be used to derive secret keys.

OT.KeyIntegrity Protection of integrity of keys

The value and critical attributes of keys (secret or public) have their integrity protected by the TOE against unauthorized modification (unauthorized modifications include making unauthorized copies of a key such that the attributes of the copy can be changed without the same authorization as for the original key). Critical attributes in this context are defined to be those implementation-level attributes of a key that could be used by an attacker to cause the equivalent of a modification to the key value by other means (e.g. including changing the cryptographic functions for which a key can be used, the users with access to the key, or the identifier of the key). This objective includes protection of the keys during generation, storage (including external storage), and use.

OT.Auth Authorisation for use of TOE functions and data

The TOE carries out an authentication/authorization check on all subjects before allowing them to use the TOE. The following types of entity are distinguished for the purposes of authorization (i.e. each type has a distinct method of authorization): administrators of the TOE, users of TOE cryptographic functions (client applications using secure channels), users of secret keys. In particular, the TOE always requires authorization before using a secret key.

OT.KeyUseConstraint Constraints on use of keys

Any key (secret or public) has an unambiguous definition of the purposes for which it can be used, in terms of the cryptographic functions or operations (e.g. encryption or signature) that it is permitted to be used for. The TOE rejects any attempt to use the key for a purpose that is not permitted. The TOE also has an unambiguous definition of the subjects that are permitted to access the key (and the purposes for which this access can be used) and allows this to be set to the granularity of an individual subject – these access constraints apply to use of the key even where the key value is not accessible. This objective means that the TOE also prevents unauthorized use of any cryptographic functions that use a key.

OT.KeyUseScope Defined scope for use of a key after authorisation

The TOE is required to define and apply clearly stated limits on when authorization and re-authorization are required in order for a secret key to be used. For example the TOE may allow secret keys to be used for a specified time period or number of uses after initial authorization, or for may allow the key to be used until authorization is explicitly rescinded. As another example, the TOE may implement a policy that requires re-authorization before every use of a secret key.

OT.DataConf Protection of confidentiality of sensitive client application data

The TOE provides secure channels to client applications that can be used to protect the confidentiality of sensitive data (such as authentication/authorization data) during transmission between the client application and the TOE, or during transmission between separate parts of the TOE where that transmission passes through an insecure environment.

OT.DataMod Protection of integrity of client application data

The TOE provides secure channels to client applications that can be used to protect the integrity of sensitive data (such as data to be signed, authentication/authorization data or public key certificates) during transmission between the client application and the TOE.

OT.ImportExport Secure import and export of keys

The TOE allows import and export of secret keys only by using a secure method that protects the confidentiality and integrity of the data during transmission – in particular, secret keys must be exported only in encrypted form (it is not sufficient to rely on properties of a secure channel to provide the protection: the key itself must be encrypted). The TOE also allows individual secret keys under its control to be identified as non-exportable, in which case any attempt to export them will be rejected

automatically. Public keys may be imported and exported in a manner that protects the integrity of the data during transmission. Assigned keys cannot be imported or exported.

OT.Backup Secure backup of user data

Any method provided by the TOE for backing up user data, including secret keys, preserves the security of the data and is controlled by authorized Administrators. The secure backup process preserves the confidentiality and integrity of the data during creation, transmission, storage and restoration of the backup data. Backups also preserve the integrity of the attributes of keys.

OT.RNG Random number quality

Random numbers generated and provided to client applications for use as keys, authentication/authorization data, or seed data for another random number generator that is used for these purposes shall meet a defined quality metric in order to ensure that random numbers are not predictable and have sufficient entropy.

OT.TamperDetect Tamper Detection

The TOE shall provide features to protect its security functions against tampering. In particular the TOE shall make any physical manipulation within the scope of the intended environment (adhering to OE.Env) detectable for the administrators of the TOE.

OT.FailureDetect Detection of TOE hardware or software failures

The TOE detects faults that would cause some other security property to be weakened or to fail, including: environmental conditions outside normal operating range (including temperature and power), failures of critical TOE hardware components (including the RNG), corruption of TOE software. On detection of a fault, the TOE takes action to maintain its security and the security of the data that it contains and controls.

OT.Audit Generation of audit trail

The TOE creates audit records for security-relevant events, recording the event details and the subject associated with the event. The TOE ensures that the audit records are protected against accidental or malicious deletion or modification of records by providing tamper protection (either prevention or detection) for the audit log.

OT.P_ACCESSIBILITY

The TOE guarantees exclusive access to the resources assigned to a partition, to the partition that has been assigned to them during the partition creation process. Each partition will have access only to its assigned resources and will not be able to access the resources assigned to another partition.

OT.P_MP_FUNCTIONALITY

The TSF shall prevent any execution of a management function reserved to Master Partition by User Partitions.

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

The security objectives for the Operational Environment determine the responsibility of the environment in countering the threats, enforcing the OSPs and upholding the assumptions. Each objective must be traced back to aspects of identified threats to be countered by the environment, to aspects of OSPs to be enforced by the environment and to assumptions to be uphold by the environment.

OE.ExternalData Protection of data outside TOE control

Where copies of data protected by the TOE are managed outside of the TOE, client applications and other entities shall provide appropriate protection for that data to a level required by the application context and the risks in the deployment environment. This includes protection of data that is exported from, or imported to, the TOE (such as audit data and encrypted keys). In particular, any backups of the TOE and its data shall be maintained in a way that ensures appropriate controls over making backups, storing backup data, and using backup data to restore an operational TOE. The number of sets of backup data shall not exceed the minimum needed to ensure continuity of the TSP service. The ability to restore a TOE to an operational state from backup data shall require at least dual person control (i.e. the participation and approval of more than one authenticated administrator).

OE.Env Protected operating environment

The TOE shall operate in a protected environment that limits physical access to the TOE to authorized Administrators. The TOE software and hardware environment (including client applications) shall be installed and maintained by Administrators in a secure state that mitigates against the specific risks applicable to the deployment environment, including (where applicable):

- protection against loss or theft of the TOE or any of its externally stored assets
- inspections to deter and detect tampering (including attempts to access side-channels, or to access connections between physically separate parts of the TOE, or parts of the hardware appliance)
- protection against the possibility of attacks based on emanations from the TOE (e.g. electromagnetic emanations) according to risks assessed for the operating environment
- protection against unauthorized software and configuration changes on the TOE and the hardware appliance
- protection to an equivalent level of all instances of the TOE holding the same assets (e.g. where a key is present as a backup in more than one instance of the TOE).

OE.DataContext Appropriate use of TOE functions

Any client application using the cryptographic functions of the TOE shall ensure that the correct data are supplied in a secure manner (including any relevant requirements for authenticity, integrity and confidentiality). For example, when creating a digital signature over a DTBS the client application shall ensure that the correct (authentic, unmodified) DTBS/R is supplied to the TOE, and shall correctly and securely manage the signature received from the TOE; and when certifying a public key the client application shall ensure that necessary checks are made to prove possession of the corresponding private key. The client application may make use of appropriate secure channels provided by the TOE to support these security requirements. Where required by the risks in the operational environment a suitable entity (possibly the client application) shall perform a check of the signature returned from the TOE, to confirm that it relates to the correct DTBS. Client applications shall be responsible for any required logging of the uses made of the TOE services, such as signing (or sealing) events. Similar requirements shall apply in local use cases where no client application need be involved, but in which the TOE and its user data (such as keys used for signatures) need to be configured in ways that will support the need for security requirements such as sole control of signing keys. Appropriate procedures shall be defined for the initial creation of data and continuing operation of the TOE according to the specific risks applicable to the deployment environment and the ways in which the TOE is used.

OE.Uauth Authentication of application users

Any client application using the cryptographic services of the TOE shall correctly and securely gather identification and authentication/authorization data from its users and securely transfer it to the TOE (protecting the confidentiality of the authentication/authorization data as required) when required to authorize the use of TOE assets and services.

OE.AuditSupport Audit data review

The audit trail generated by the TOE will be collected, maintained and reviewed by a System Auditor according to a defined audit procedure for the TSP.

OE.AppSupport Application security support

Procedures to ensure the ongoing security of client applications and their data shall be defined and followed in the environment, and reflected in use of the appropriate TOE cryptographic functions and parameters, and appropriate management and administration actions on the TOE. This includes, for example, any relevant policies on algorithms, key generation methods, key lengths, key access, key import/export, key usage limitations, key activation, cryptoperiods and key renewal, and key/certificate revocation.

4.3 SECURITY OBJECTIVES RATIONALE

The following table provides a mapping of security objectives tracing each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective, and each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective. This illustrates that the security objectives counter all threats, the security objectives enforce all OSPs and the security objectives for the operational environment uphold all assumptions.

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OT.P_ACCESSIBILITY	OT.P_MP_FUNCTIONALITY	OE.ExternalData	OE.Env	OE.DataContext	OE.Uauth	OE.AuditSupport	OE.AppSupport
T.KeyDisclose	X		X				X		X	X		X					X	X				
T.KeyDerive		X									X											
T.KeyMod			X						X	X		X						X				
T.KeyMisuse				X	X																	
T.KeyOveruse						X																
T.DataDisclose							X												X			X
T.DataMod								X											X			X
T.Malfunction													X									
T.P_CROSS_ACCESS															X							
T.P_EXECUTION																X						
P.Algorithms		X																				
P.KeyControl	X	X		X	X	X			X	X												

	OT.PlainKeyConf	OT.Algorithms	OT.KeyIntegrity	OT.Auth	OT.KeyUseConstraint	OT.KeyUseScope	OT.DataConf	OT.DataMod	OT.ImportExport	OT.Backup	OT.RNG	OT.TamperDetect	OT.FailureDetect	OT.Audit	OT.P_ACCESSIBILITY	OT.P_MP_FUNCTIONALITY	OE.ExternalData	OE.Env	OE.DataContext	OE.Uauth	OE.AuditSupport	OE.AppSupport
P.RNG											X											
P.Audit														**X**								
A.ExternalData																	**X**					
A.Env																		**X**				
A.DataContext																			**X**			
A.Uauth																				**X**		
A.AuditSupport																					**X**	
A.AppSupport																						**X**

Table 4 Security Objectives vs Security Problem Definition

THREATS

T.KeyDisclose: T.KeyDisclose is addressed by the requirement in **OT.PlainKeyConf** to keep plaintext secret keys unavailable, and this is supported in terms of controls over key attributes (which might threaten the confidentiality of the key if modified) in **OT.KeyIntegrity**. The confidentiality of secret keys that are exported is protected partly by the use of a secure channel as described in **OT.DataConf** and the requirements for import and export in **OT.ImportExport** (including the requirement to export secret keys only in encrypted form, or to be able to exclude the export of a key entirely). Physical tamper protection of the keys is provided by **OT.TamperDetect** (supported by an appropriate inspection procedure as required in **OE.Env**). Protection of secret key confidentiality during backup is ensured by **OT.Backup**. The environment also contributes to maintaining secret key confidentiality by protecting any versions of a secret key that may exist outside the **[TOE]**, as in **OE.ExternalData**, and by protecting the operation of the **[TOE]** itself by providing a secure environment, as in **OE.Env**.

T.KeyDerive: T.KeyDerive is addressed by the choice of algorithms that have been endorsed for the appropriate purposes, and this is described in **OT.Algorithms**. Where keys are generated by the **[TOE]** then the use of a suitable random number generator is required by **OT.RNG** in order to mitigate the risk that an attacker can guess or deduce the key value.

T.KeyMod: T.KeyMod is addressed by requiring integrity protection of secret and public keys, and their critical attributes in **OT.KeyIntegrity**, and by requiring use of secure channels that protect integrity if a key is imported or exported (**OT.ImportExport**). Protection of key integrity during backup is ensured by **OT.Backup**. Physical tamper protection of the keys is provided by **OT.TamperDetect** (supported by an appropriate inspection procedure as required in **OE.Env**).

T.KeyMisuse: T.KeyMisuse raises the possibility of a secret key being used for an unintended and unauthorised purpose, and is addressed by the requirement in **OT.Auth** for the **[TOE]** to carry out an authorisation check before using a secret key. **OT.KeyUseConstraint** expands on this to set out requirements for the granularity of authorisation.

T.KeyOverUse: T.KeyOveruse is concerned with the possibility that more uses may be made of an authorised key than were intended, and this is addressed by the requirements of **OT.KeyUseScope** which requires controls to be specified and enforced for any re-authorisation conditions that the **[TOE]** allows a user to define.

T.DataDisclose: T.DataDisclose is concerned with the transmission of data between client applications and the **[TOE]**, or between separate parts of the **[TOE]** where the transmission passes through an insecure environment. This is addressed by **OT.DataConf**, which requires the **[TOE]** to provide secure channels to protect such communications. The appropriate use of such channels is a requirement for the environment as expressed in **OE.DataContext**, as is the use of appropriate procedures in **OE.AppSupport**.

T.DataMod: T.DataMod is concerned with the possibility of unauthorised modification of data transmitted between a client application and the **[TOE]**, and this is addressed by **OT.DataMod** which requires that the **[TOE]** provides secure channels that can be used to protect the integrity of data that they carry. As with T.DataDisclose, the appropriate use of such channels is a requirement for the environment as expressed in **OE.DataContext**, as is the use of appropriate procedures in **OE.AppSupport**.

T.Malfunction: T.Malfunction is addressed by the requirement in **OT.FailureDetect** for the **[TOE]** to detect certain types of fault.

T.P_CROSS_ACCESS: The security objective **OT.P_ACCESSIBILITY** addresses this threat by ensuring the exclusive access to the resources assigned to each partition to the partition owning the resources.

T.P_EXECUTION: The security objective **OT.P_MP_FUNCTIONALITY** addresses this threat by preventing the execution of management functions reserved to Master Partition by User Partitions

The following table maps the threats of the security problem established to the security objectives of the TOE and the security objectives of the operational environment.

Threats	Security Objectives
T.KeyDisclose	OT.PlainKeyConf OT.KeyIntegrity OT.DataConf OT.ImportExport OT.TamperDetect OT.Backup OE.Env OE.ExternalData
T.KeyDerive	OT.Algorithms OT.RNG
T.KeyMod	OT.KeyIntegrity OT.ImportExport OT.Backup OT.TamperDetect OE.Env
T.KeyMisuse	OT.Auth OT.KeyUseConstraint
T.KeyOveruse	OT.KeyUseScope
T.DataDisclose	OT.DataConf OE.DataContext OE.AppSupport
T.DataMod	OT.DataMod OE.DataContext OE.AppSupport
T.Malfunction	OT.FailureDetect
T.P_CROSS_ACCESS	OT.P_ACCESSIBILITY
T.P_EXECUTION	OT.P_MP_FUNCTIONALITY

Table 5 Threats vs Security Objectives

ORGANIZATIONAL SECURITY POLICIES

P.Algorithms: P.Algorithms requires the use of key generation and other cryptographic functions that are endorsed by appropriate authorities, and this is addressed by **OT.Algorithms**.

P.KeyControl: P.KeyControl requires that the **[TOE]** can provide controls and support a key lifecycle to ensure that secret keys can be reliably protected against use by those other than the owner of the key, and that the keys can be confined to use for certain cryptographic functions. This is addressed by a combination of **[TOE]** objectives as follows:

OT.PlainKeyConf protects the value of the secret key to prevent the possibility of it being used by unauthorized subjects.

OT.Algorithms ensures that endorsed algorithms that employ and support suitable properties and procedures are provided by the **[TOE]**.

OT.Auth, **OT.KeyUseConstraint** and **OT.KeyUseScope** ensure that the **[TOE]** can provide well-defined limits on the use of a key when it is authorised (as described above for T.KeyMisuse and T.KeyOveruse).

OT.ImportExport and **OT.Backup** ensure protection of keys when they are transmitted outside the **[TOE]** to client applications or for backup purposes, including the prevention of export of Assigned Keys.

P.RNG: P.RNG is directly enforced by **OT.RNG**.

P.Audit: P.Audit requires the **[TOE]** to provide an audit trail and this is addressed directly by **OT.Audit** (which includes protection of the audit records).

The following table maps the organizational security policies of the problem established to the security objectives of the TOE and the security objectives of the operational environment.

OSPs	Security Objectives
P.Algorithms	OT.Algorithms
P.KeyControl	OT.PlainKeyConf OT.Algorithms OT.Auth OT.KeyUseConstraint OT.KeyUseScope OT.ImportExport OT.Backup
P.RNG	OT.RNG
P.Audit	OT.Audit

Table 6 OSPs vs Security Objectives

ASSUMPTIONS

Each of the Assumptions in section 3.5 is directly matched by a security objective for the operational environment in section 4.2. The wording of each objective for the operational environment includes the wording of each assumption, and no further rationale is therefore given here.

5 SECURITY REQUIREMENTS

This section defines the Security functional requirements (SFRs) and the Security assurance requirements (SARs) that fulfill the TOE.

The following convention has been used regarding the claiming of this security target with **[CEN EN 419221-5]:**

1. For those SFRs which are included in **[CEN EN 419221-5]**, the solved operations in the PP have been written in this ST by copying the final result of the operation using the same typography. The unsolved operations follows the convention specified in (2) below.
2. For those SFRs which are not included in **[CEN EN 419221-5]** and are from **[CCPART2]**, the following convention has been used:
    - Assignments. They appear between square brackets. The word "assignment" is maintained and the resolution is presented in ***boldface, italic and blue color.***
    - Selections. They appear between square brackets. The word "selection" is maintained and the resolution is presented in ***boldface, italic and blue color.***
    - Iterations. It includes "/" and an "identifier" following requirement identifier that allows to distinguish the iterations of the requirement. Example: FCS_COP.1/XXX.
    - Refinements: the text where the refinement has been done is shown ***bold, italic, and light red color.*** Where part of the content of a SFR component has been removed, the removed text is shown in ~~***bold, italic, light red color and crossed out.***~~

5.1 SECURITY FUNCTIONAL REQUIREMENTS

FCS: CRYPTOGRAPHIC SUPPORT

FCS_CKM.1: Cryptographic key generation – Key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***[assignment: specified in Table 7]*** and specified cryptographic key sizes ***[assignment: specified in Table 7]*** that meet the following: ***[assignment: specified in Table 7]***.

Key Generation Algorithm	Key Sizes	Applicable Standards
ECC Key Generation	NIST Curves: P-224, P-256, P-384, P-521	[FIPS 186-5] Appendix A.2
	Brainpool curves:	RFC- 5639 [IETF RFC Brainpool]

	BrainpoolP224r1, BrainpoolP256r1, BrainpoolP384r1, BrainpoolP512r1	
	Koblitz Curves: Secp256K1	SP 800-186 Appendix H.2
RSA Key Generation	Modulus length 2048 to 4096 with increment of 256 bits	Chapter 5.1 from [FIPS 186-5]
AES Key Generation	128, 192 and 256 bits	Chapter 10.1.1 (Hash DRBG) from [SP 800-90Ar1]
Generic secret (HMAC) key generation	8 to 6400 bits	Chapter 10.1.1 (Hash DRBG) from [SP 800-90Ar1]
PQC ML-KEM Key Generation	512/768/1024	Chapter 7.1 from [NIST.FIPS.203]
PQC-ML_DSA Key generation	44, 65, 87	Chapter 5.1 and 6.1 from [NIST.FIPS.204]

Table 7 Key generation support table

Application Note

Key generation is linked to the setting of security attributes of a key (including the link to a subject who owns the key, via the setting of authorization data) as in FMT_MSA.1/GenKeys and FMT_MSA.1/Akeys.

The internal RNG of the TOE is used in the key generation process.

FCS_CKM.5: Cryptographic key Derivation

FCS_CKM.5.1 The TSF shall derive cryptographic keys ***[assignment: key type (AES, HMAC secret keys)]*** *from* ***[assignment: input parameters specified in Table 8]*** in accordance with a specified cryptographic key derivation algorithm ***[assignment: specified in Table 8]*** and specified cryptographic key sizes ***[assignment: specified in Table 8]*** that meet the following: ***[assignment: specified in Table 8]***.

Key Type	Input Parameters	Key Derivation Algorithm	Key Sizes	Supported PRF / Hashing Function / Cipher	Applicable Standards
KDF-CTR	Key derivation key (AES or HMAC key as base key), Label, context and Length of derived keying material	Counter Mode KDF	128,192 and 256 bits when AES is cipher. 128 – 512 bits when HMAC PRF is used.	AES-CMAC, HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA512.	Counter Mode KDF from [SP 800-108r1-upd1] Revision 1 (Section 4.1) [KDF in counter mode]
KDF-Hash	Hash type, Shared secret (After ANSI X9.63 KDF) and salt	ANSI X9.63 KDF	256, 384, and 512, dependent on hashing function.	SHA256, SHA384, and SHA512	Key Derivation Functions in [SP 800-135r1] and [ANSIX9.63]
KDF-Hash	Hash type, Shared secret (After ECDH) and salt	ECDH	P-224, P-256, P-384, and P521	Not applicable	ECC - Ephemeral Unified, One step KDF (Chapter 4) from [SP800-56Cr2]
HMAC-KDF	Password, salt, iteration count and length of the derived key	PBKDF	256 bits	HMAC-SHA256	[SP800-132]
KDF-HMAC/CMAC	Extract: hash function, Salt, key Expand: a pseudorandom key (PRK, output of extract operation), context	Two Step KDF	256, 384 and 512 depends on hashing function	HMAC/CMAC KDFs	Chapter 5 from [SP800-56Cr2]

Key Type	Input Parameters	Key Derivation Algorithm	Key Sizes	Supported PRF / Hashing Function / Cipher	Applicable Standards
	(optional) and Length of output keying material.				

Table 8 Key derivation support table

FCS_CKM.6: Timing and event of cryptographic key destruction

FCS_CKM.6.1 The TSF shall destroy ***[assignment: specified cryptographic keys (including keying material)]*** when ***[selection: no longer needed, [assignment: other circumstances for key or keying material destruction specified in Table 9]].***

FCS_CKM.6.2

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method ***[assignment: zeroization method]*** that meets the following: ***[assignment: actions specified in Table 9].***

Application note:

The key destruction mechanism of the TSF does not imply the destruction of every single key in the TOE. It destroys keys used which are used to encrypt other keys in order to make the encrypted ones unavailable and virtually destroyed.

The destruction of assets will take place by the mechanisms and in the conditions specified in the following table:

Zeroization event		Description
HSM level	HSM zeroization	Partition deletion of all partitions in the HSM, including the Master Partition. The MCO and vendor-loaded support keys are not deleted. All PCO-loaded support keys and partition user keys are erased.
	HSM factory reset	Performs HSM zeroization and additionally erase MCO-loaded support keys, including the MCO Recovery Key.
	Vendor zeroization	This will perform an HSM factory reset and additionally erase vendor-loaded support keys. Vendor zeroize can be triggered through commands or tamper zeroize events provided recovery key is loaded.
	Tamper zeroization	Erases MCU_DEK stored on MCU, which is like “HSM Factory Reset”. This means that the

Zeroization event		Description
		Recovery Key stored on MCU is crypto erased and read operation will return garbage value.
Partition level	Partition zeroization	Erases all user keys (that are created by partition users) and the Part_DEK to ensure that all other partition keys are unrecoverable.
	Partition factory reset	Performs a zeroization of the partition support keys and user keys listed at section 6.2 except the PID and PID_Cert.
	Partition delete	Equivalent to partition factory reset plus the partition instance is also teared down or removed and should be create again if intended to re-use.
DRBG Internal State	DRBG access	The DRBG maintains an internal cache, which is cleared (zeroized) each time it is accessed—such as during key or IV generation etc.,—ensuring that any used data of specified lengths is securely erased.

Table 9 Key destruction support table

FCS_COP.1/Symmetric: Cryptographic operation – Symmetric encryption & decryption

FCS_COP.1.1/Symmetric The TSF shall perform ***[assignment: symmetric encryption and decryption]*** in accordance with a specified cryptographic algorithm ***[assignment: specified in Table 10]*** and cryptographic key sizes ***[assignment: specified in Table 10]*** that meet the following: ***[assignment: specified in Table 10].***

Algorithm	Key Sizes	Operation Mode	Applicable Standards
AES	128, 192 and 256 bits	CBC	[FIPS 197-upd1] chapter 5 and [SP800-38A] chapter 6
AES	128, 192 and 256 bits	AES-KW and AES-KWP	[FIPS 197-upd1] chapter 5 and [SP800-38F] chapter 6
AES	128, 192 and 256 bits	CCM	[SP800-38C] chapter 6
AES	128, 192 and 256 bits	GCM	[SP800-38D] chapter 6

Table 10 Symmetric cryptography support table

FCS_COP.1/MAC: Cryptographic operation – Message Authentication Code

FCS_COP.1.1/MAC The TSF shall perform ***[assignment: message authentication code generation and verification]*** in accordance with a specified cryptographic algorithm ***[assignment: specified in Table 11]*** and cryptographic key sizes ***[assignment: specified in Table 11]*** that meet the following: ***[assignment: specified in Table 11]***.

MAC Algorithm	Key Sizes (bits)	Supported PRF / Hashing Function	Applicable Standards
AES-CMAC	128, 192 and 256	AES	[SP800-38B] chapter 6
HMAC	8 <= k <= 6400	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3- 224, SHA3-256, SHA3-384, SHA3-512	[HMAC] and [FIPS 198-1]
AES-GMAC	128, 192 and 256	AES	[SP800-38D] chapter 7

Table 11 Message Authentication Code support table

FCS_COP.1/Asymmetric: Cryptographic operation – Asymmetric cryptography

FCS_COP.1.1/Asymmetric The TSF shall perform ***[assignment: asymmetric encryption, decryption, signature generation and verification]*** in accordance with a specified cryptographic algorithm ***[assignment: specified in Table 12]*** and cryptographic key sizes ***[assignment: specified in Table 12]*** that meet the following: ***[assignment: specified in Table 12]***.

Algorithm	Key Sizes	Hash Algorithm	Applicable Standards
RSA encryption and decryption	Modulus length 2048 to 4096 with increment of 256 bits	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA3- 224, SHA3-256, SHA3-384, SHA3-512	RSAES-OAEP from [PKCS#1]
ECDSA signature generation and verification	NIST curves: P-224, P-256, P-384, P-521 Brainpool curves: BrainpoolP224r1, BrainpoolP256r1, BrainpoolP384r1, BrainpoolP512r1 Koblitz Curves: Secp256K1	SHA-224, 256, 384, and 512	[FIPS 186-5] chapter 6 and [SP800-186] chapter 3.1.2

Algorithm	Key Sizes	Hash Algorithm	Applicable Standards
RSA signature generation and verification	Modulus length 2048 to 4096 with increment of 256 bits	SHA-224, 256, 384 and 512	RSASSA-PSS and RSASSA-PKCS1-v1_5 from [PKCS#1] chapters 8.1.1 and 8.2.1 and chapter 5 from [FIPS 186-5]
PQC-ML-KEM Encap and Decap	512, 768, 1024	SHAKE128 and SHAKE256	Chapter 7.2 and 7.3 from [NIST.FIPS.203]
PQC-ML_DSA Signature generation and verification	44, 65, 87	SHAKE128 and SHAKE256	Chapter 6.2 and 6.3 from [NIST.FIPS.204]

Table 12 Asymmetric cryptography support table

FCS_COP.1/Digest: Cryptographic operation – Digest

FCS_COP.1.1/Digest The TSF shall perform ***[assignment: digest]*** in accordance with a specified cryptographic algorithm ***[assignment: specified in Table 13]*** and cryptographic key sizes ***[assignment: none]*** that meet the following: ***[assignment: specified in Table 13]***.

Message Digest Algorithm	Applicable Standards
SHA-1	[FIPS 180-4] chapter 6.1
SHA-256	[FIPS 180-4] chapter 6.2
SHA-384	[FIPS 180-4] chapter 6.5
SHA-512	[FIPS 180-4] chapter 6.4
SHA3-256	[FIPS 202] chapter 5.2 and 6.1
SHA3-384	[FIPS 202] chapter 5.2 and 6.1
SHA3-512	[FIPS 202] chapter 5.2 and 6.1
SHAKE128	[FIPS 202] chapter 6.2 and 6.3
SHAKE256	[FIPS 202] chapter 6.2 and 6.3

Table 13 Message digest support table

FCS_RBG.1: Random Bit Generation

The FCS_RBG.1 Random bit generation (RBG) requires random bit generation to be performed in accordance with selected standards. It also specifies whether the initial seeding is done via an internal or external noise source, as well as when and how an RBG's state is updated.

FCS_RBG.1.1 The TSF shall perform deterministic random bit generation services using ***[assignment: Hash_DRBG and CTR_DRBG algorithms***] in accordance with ***[assignment: NIST SP 800-90Ar1, section 10.1.1 and NIST SP 800-90Ar1, section 10.2]*** after initialization.

FCS_RBG.1.2 The TSF shall use a ***[selection: TSF entropy source [assignment: TRNG], TSF interface for obtaining entropy]*** for initialization and reseeding.

FCS_RBG.1.3 The TSF shall update the DRBG state by ***[selection: reseeding]*** using a ***[selection: TSF entropy source [assignment: TRNG], TSF interface for obtaining entropy [assignment : OCTEON HW RBG]*** in the following situations***: [selection:***

— on demand;

in accordance with ***[assignment: SP800-90A]***

FCS_RBG.3 Random bit generation (internal seeding – single source)

FCS_RBG.3.1

The TSF shall be able to seed the DRBG using a ***[selection: choose one of: TSF hardware-based entropy source] [assignment: OCTEON HW RBG]*** with ***[assignment: 12672]*** bits of min-entropy.

FCS_RBG.6 Random bit generation service

FCS_RBG.6.1

The TSF shall provide a ***[selection: software]*** interface to make the DRBG output, as specified in FCS_RBG.1 Random bit generation (RBG), available as a service to entities outside of the TOE.

FCS_RNG: Generation of random numbers

FCS_RNG.1.1 The TSF shall provide a ***[selection: physical]*** random number generator that implements ***[assignment: TRNG (True Random Number Generator)) conformant to NIST SP 800-90B [SP800-90B] (OCTEON HW RBG)]***.

FCS_RNG.1.2 The TSF shall provide ***[selection: octets of bits]*** that meet ***[assignment: OCTEON HW RBG produces 2.673 bits of minimum entropy in each 128 bits of output.]***

FIA: IDENTIFICATION AND AUTHENTICATION

FIA_AFL.1: Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when ***[selection: an administrator configurable positive integer within [assignment: specified in Table 14]]*** unsuccessful authentication or authorization attempts occur related to *consecutive failed authentication or authorization attempts*.

Role	Number of consecutive authentication/authorization failures	Functionality being blocked	Unblocking condition
MCO	A positive integer within the range [1 to 20], configurable by MCO.	All HSM functionality. (Only MCO operations to manage HSM will be blocked)	MCO can be unblocked with the AOTAC private key, get the challenge (without MCO login) after MCO is locked -> Sign the challenge using the AOTAC private key and run unlockCO. If hsm_audit_log flag is enabled, when the host is rebooted or the PF driver reloaded, the firmware will automatically zeroize. If hsm_audit_log flag is disabled, A zeroizeHSM command can be issued and new credentials can be provided as part of the re-init command.
PCO	A positive integer within the range [1 to 20], configurable by MCO per partition.	All the functionalities of the related partition	PCO can be unblocked with POTAC private key, get the challenge (without PCO login) after PCO is locked -> Sign the challenge using the POTAC private key and run unlockCO.
PCU	Same number as for the corresponding PCO	Partition Crypto User user is locked out while the rest of users of the partition are still operational.	PCO can unblock PCU
AU	Same number as for the corresponding PCO	Appliance User user login and related capabilities	PCO or MCO can unblock AU
PRE-CO	Same number as for the corresponding PCO	All the functionalities of the partition is blocked till PRECO is unlocked and password is changed	Partition user can zeroizeHSM and initialize with new preCO credentials

Table 14 Authentication failure support table

FIA_AFL.1.2 When the defined number of unsuccessful authentication **or authorisation** attempts has been ***[selection: met]***, the TSF shall *block access to* ***[assignment: specified in Table 14]*** *until* ***[selection: unblocked by [assignment: specified in Table 14]]***.

Application Note

zeorizeHSM (partition) includes clearing the logs and removing the existing PAK/PAC and masking key and regenerating new ones for the current FIPS state (the POTAC is retained).

FIA_UAU.1: Timing of authentication – Roles Authentication

FIA_UAU.1.1 The TSF shall allow

(1) Self-test according to FPT_TST.1,

(2) Identification of the user by means of TSF required by FIA_UID.1

(3) [assignment:

- ***Execution of following operations that can be performed without login that do not compromise TOE assets:***
  - ***Get the partition or all partitions information***
  - ***Get the HSM information - Firmware build version, ToE Model number, bootloader version, session information, FIPS state, Fingerprint of vendor and MCO, HSM time, session count, Available and occupied TOE memory, CO failure count, MCU sensors status, MCU tamper protection information and session count.***
  - ***Get the Firmware version.***
  - ***Application initialization – Registers the application with HSM.***
  - ***Application finalization – Unregisters the applications with HSM and also cleanup of all resources allocated during the application initialization.***
  - ***Open a session.***
  - ***Close a session.***
  - ***Closing of all sessions.***
  - ***Getting the session info.***
  - ***Getting the key attribute size.***
  - ***Getting the public key attribute value.***
  - ***Getting all key attributes size.***
  - ***Getting all public key attributes values.***
  - ***Zeroization of HSM when any HSM component is compromised or tamper event detected.***
  - ***Get a challenge to be signed by either HSM/Partitions owner to move the session to "unlocked" state using "unlockco" command.***
  - ***Get the list of users.***
  - ***Get login failure count.***
  - ***Get HSM Diagnostic stats.***
  - ***Get HSM Diagnostic information.***
  - ***Get the policies set.***
  - ***Get M Value used during the MofN.***
  - ***Get a HSM label name.***
  - ***Get unlinked (not deleted) objects.***
  - ***Unlock STM lock.***

    - ***Perform all certification authentication operations except store and remove of HSM certs.***
    - ***Find all public keys.***
    - ***Get the RTC and system time from the HSM***
    - ***Get basic cluster info (get key hash(es), set Node ID)***
    - ***Run FIPS tests using command line.***
    - ***Get information of HUK from NOR.***
- ***Establishment of a secure TLS channel]***

on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.6/KeyAuth: Re-authenticating

FIA_UAU.6.1/KEYAUTH The TSF shall **authorise and re-authorise** the user **for access to a secret key** under the conditions

(1) Authorisation in order to be granted initial access to the key; and

(2) ***[selection:***

- ***Re-authorisation of [assignment: both General Keys and Assigned keys] under the following conditions: [selection:***
    - ***after explicit rescinding of previous authorization for access to the secret key]];***

Application Note

It is the responsibility of the client application to make appropriate use of any re-authentication conditions according to the application context (cf. OE.DataContext and OE.AppSupport).

FIA_UID.1: Timing of identification – Roles Identification

FIA_UID.1.1 The TSF shall allow

(1) Self test according to FPT_TST.1

(2) [assignment: those described in FIA_UAU.1]

on behalf of the user to be performed before the user is identified.

Application Note

Local Identification is only way for accessing Master Partition.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FDP: USER DATA PROTECTION

FDP_ACC.1/KeyUsage: Subset access control

FDP_ACC.1.1/KeyUsage The TSF shall enforce the *Key Usage SFP* on

(1) subjects: all

(2) objects: keys

(3) operations: all.

FDP_ACC.1/Backup: Subset access control

FDP_ACC.1.1/Backup The TSF shall enforce the *Backup SFP* on

(1) subjects: all

(2) objects: keys

(3) operations: backup, restore.

Application Note

The relevant security requirements are trivially met because no backup facility is provided for the whole TSF. However, the backup facility is available for each logical partition.

FDP_ACF.1/KeyUsage: Security attribute based access control

FDP_ACF.1.1/KeyUsage The TSF shall enforce the *Key Usage SFP* to objects based on the following:

(1) whether the subject is currently authorized to use the secret key

(2) whether the subject is currently authorised to change the attributes of the secret key

(3) the cryptographic function that is attempting to use the secret key.

Application Note

Whether a subject is currently authorised for access to a secret key is determined by whether the subject has submitted the correct authorisation data for the key, and whether this authorisation is yet subject to one or more of the re-authorisation conditions in FIA_UAU.6/KeyAuth.

Whether a subject is currently authorised to change the attributes of a secret key is determined by the iterations of FMT_MSA.1

FDP_ACF.1.2/KeyUsage The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) Attributes of a key shall only be changed by an authorised subject, and only as permitted in Table 15 for General keys and Table 16 for Assigned keys.

(2) Only subjects with current authorisation for a specific secret key shall be allowed to carry out operations using the plaintext value of that key.

(3) Only cryptographic functions permitted by the secret key's Key Usage attribute shall be carried out using the secret key.

FDP_ACF.1.3/KeyUsage The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none.*

FDP_ACF.1.4/KeyUsage The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none.*

Application Note (1)

The requirements of FDP_ACF.1/KeyUsage apply regardless of how the key is stored by the TOE, including when the key is externally stored

Application Note (2)

FDP_ACF.1.2/KeyUsage (1) refers to controls over changing attributes that are specified in more detail in the iterations of FMT_MSA.1.

FDP_ACF.1.2/KeyUsage (2) requires that a key can only be used when the relevant subject has been authorised either by presenting the correct authorisation data for the key as part of the request for the operation or else the authorisation has previously been presented by the subject and the current use of the key does not yet require re-authorisation according to FIA_UAU.6/KeyAuth (meaning that the current usage is therefore within the usage constraints for time and number of uses since the last authorisation of use of the key). The reference to use of the plaintext value of the key does not imply that a subject has access to that value, only that it can be used to carry out operations within the TOE – references to operations of this sort are thus distinguished from operations that may use an encrypted form of a secret key (e.g. for external storage of keys) and that are not necessarily restricted in this way.

FDP_ACF.1/Backup: Security attribute based access control

FDP_ACF.1.1/Backup The TSF shall enforce the *Backup SFP* to objects based on the following:

(1) whether the subject is an administrator.

FDP_ACF.1.2/Backup The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) Only authorised administrators shall be able to perform any backup operation provided by the TSF to create backups of the TSF state or to restore the TSF state from a backup

(2) Any restore of the TSF shall only be possible under at least dual person control, with each person being an administrator

(3) Any backup and restore shall preserve the confidentiality and integrity of the secret keys, and the integrity of public keys

(4) Any backup and restore operations shall preserve the integrity of the key attributes, and the binding of each set of attributes to its key.

FDP_ACF.1.3/Backup The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*.

FDP_ACF.1.4/Backup The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*.

Application Note (1)

Preserving the binding of a set of attributes to its key (in FDP_ACF.1.2/Backup (4)) means that it is not possible for the attributes to be changed during a backup operation, or by modification of the backup data while it is away from the TSF.

Backups may contain keys whose export flag attribute marks them as 'non-exportable'.

Application Note (2)

The relevant security requirements are trivially met because no backup facility is provided for the whole TSF. However, the backup facility is available for each logical partition.

FDP_IFC.1/KeyBasics: Subset information flow control

FDP_IFC.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP* on

(1) subjects: all

(2) information: keys

(3) operations: all.

FDP_IFC.1/Partitions: Complete information flow control– Partition Flow Control

FDP_IFC.1.1/Partitions The TSF shall enforce the ***[assignment: Partitions flow control SFP]*** on ***[assignment***

(1) subjects: partitions

(2) information: user data

(3) operations

- access CPT assigned hardware queue

- access RAM memory assigned partition].

FDP_IFF.1/KeyBasics: Simple security attributes

FDP_IFF.1.1/KeyBasics The TSF shall enforce the *Key Basics SFP* based on the following types of subject and information security attributes:

(1) whether a key is a secret or a public key

(2) whether a secret key is an Assigned Key

(3) whether channels selected to export keys are secure

(4) the value of the Export Flag of a key.

Application Note

TOE keys do not include an attribute that models all the characteristics of the "Assigned" attribute. However, all the behavior expected for the "Assigned" attribute is modeled by the TOE through the following attributes: NEVER_EXTRACTABLE, LOCAL and MODIFIABLE. LOCAL is never modifiable, MODIFIABLE cannot return to "true" once it is set to false" and NEVER_EXTRACTABLE cannot be set to "false" once is set to "true"

FDP_IFF.1.2/KeyBasics The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

(1) Export of secret keys shall only be allowed provided that the secret key is not an Assigned Key, that the secret key is encrypted, and that a secure channel (providing authentication and integrity protection) is used for the export

(2) Public keys shall always be exported with integrity protection of their key value and attributes

(3) Keys shall only be imported over a secure channel (providing authentication and integrity protection)

(4) A secret key can only be imported if it is a non-Assigned key

(5) Secret keys shall only be imported in encrypted form or using split-knowledge procedures requiring at least two key components to reconstruct the key, with key components supplied by at least two separately authenticated users

(6) Unblocking access to a key shall not allow any subject other than those authorised to access the key at the time when it was blocked.

FDP_IFF.1.3/KeyBasics The TSF shall enforce the following additional information flow control rules: *none*.

FDP_IFF.1.4/KeyBasics The TSF shall explicitly authorise an information flow based on the following rules: *none*.

FDP_IFF.1.5/KeyBasics The TSF shall explicitly deny an information flow based on the following rules:

(1) No subject shall be allowed to access the plaintext value of any secret key directly

(2) No subject shall be allowed to export a secret key in plaintext.

(3) No subject shall be allowed to export an Assigned Key

(4) No subject shall be allowed to export a secret key without submitting the correct authorisation data for the key

(5) No subject shall be allowed to access intermediate values in any operation that uses a secret key

(6) A key with an Export Flag value marking it as non-exportable shall not be exported.

Application Note

A secure channel for export of keys in FDP_IFF.1.2/KeyBasics (1) or for import of keys in FDP_IFF.1.2/KeyBasics (3) is one that meets the requirements of FTP_TRP.1/Local or FTP_TRP.1/External.

The encrypted form required for keys imported or exported over a secure channel requires encryption of the key itself, in addition to any encryption provided by the secure channel.

Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) is intended only to restore the ability of subjects to authorise for access to a key by presenting the correct authorisation data. As noted for FMT_MTD.1/Unblock, the subject who unblocks the key must not be able also to use the key as a result of the unblocking (unless of course they are able to supply the correct authorisation data). This is a part of ensuring that sole control of secret keys can be achieved.

FDP_IFF.1/Partitions: Simple security attributes – Partition Flow Control

FDP_IFF.1.1/Partitions The TSF shall enforce the ***[assignment: Partitions flow control SFP]*** based on the following types of subject and information security attributes: ***[assignment: PCIe's VF's and PF (described below), partition IDs, Cgroup policies]***.

The Physical Function (PF) is a part of a network adapter that uses PCI Express (PCIe) to support single root I/O virtualization (SR-IOV). It includes special features in its PCIe settings to manage SR-IOV, allowing virtualization and the creation of PCIe Virtual Functions (VFs).

Virtual Function (VF) is a lightweight PCIe function on a network adapter that supports single root I/O virtualization (SR-IOV). The VF is associated with the PCIe Physical Function (PF) on the network adapter and represents a virtualized instance of the network adapter. Each VF has its own PCI Configuration space. Each VF also shares one or more physical resources on the network adapter, such as an external network port, with the PF and other VFs.

FDP_IFF.1.2/Partitions The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: ***[assignment:***

(1) a partition can only make use of the hardware queue of the CPT assigned to it based on PCIe's VF's and PF, partition ID

(2) a partition can only make use of the assigned RAM partition assigned to it based on PCIe's VF's and PF, partition ID, Cgroup policy].

FDP_IFF.1.3/Partitions The TSF shall enforce the ***[assignment: none]***.

FDP_IFF.1.4/Partitions The TSF shall explicitly authorise an information flow based on the following rules: ***[assignment: only the PF shall have access to the part of the firmware that provides the Master Partition functionality]***.

FDP_IFF.1.5/Partitions The TSF shall explicitly deny an information flow based on the following rules: ***[assignment: no subject shall be allowed to access other subject's information].***

FDP_RIP.1: Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* the following objects:

- *Authorization data*
- *Secret keys.*

FDP_SDI.2: Stored data integrity monitoring and action

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors* on all **keys (including security attributes)**, based on the following attributes *integrity protection data*.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall

(1) prohibit the use of the altered data

(2) notify the error to the user.

Application Note

The integrity protection data in FDP_SDI.2.1, which is referred at FMT_MSA.1/GenKeys and FMT_MSA.1/Akeys, and protects the value of the key and of its other security attributes, including when the key is externally stored by the TOE is given by key's security attributes: KCV, EKCV and hash of the key and key attributes, automatically generated when the key is created.

FTP: TRUSTED PATH/CHANNELS

FTP_TRP.1/Local: Trusted path

FTP_TRP.1.1/Local The TSF shall provide a communication path between itself and *local* **client applications** that is logically distinct from other communication paths and provides assured **authentication** of its end points and protection of the communicated data from *modification and disclosure*.

FTP_TRP.1.2/Local The TSF shall permit ***[selection: local client applications]*** to initiate communication via the trusted path.

FTP_TRP.1.3/Local The TSF shall require the use of the trusted path for ***[assignment:***

- ***initial user authentication***
- ***Master Partition management]***.

Application Note

Local client applications are located within the physical boundary of the same hardware appliance. Therefore, the local trusted path is mapped to the physical configuration, and no additional authentication or cryptographic protection are required (because of the physical security assumed in the appliance environment).

FTP_TRP.1/External: Trusted path

FTP_TRP.1.1/External The TSF shall provide a communication path between itself *remote* **external client applications** that is logically distinct from other communication paths and provides assured **authentication** of its end points and protection of the communicated data from *modification and disclosure*.

FTP_TRP.1.2/External The TSF shall permit ***[selection: remote external client applications]*** to initiate communication via the trusted path.

FTP_TRP.1.3/External The TSF shall require the use of the trusted path for ***[assignment:***

- ***initial user authentication***
- ***Management of Non-Privileged Partitions]***.

FPT: PROTECTION OF THE TSF

FPT_FLS.1: Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

(1) Self-test according to FPT_TST.1 fails

(2) Environmental conditions are outside normal operating range (including temperature and power)

(3) Failures of critical TOE hardware components (including the RNG) occur

(4) Corruption of TOE software occurs

(5) ***[assignment: none]***.

FPT_PHP.1: Passive detection of physical attack

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FPT_PHP.3: Resistance to physical attack

FPT_PHP.3.1 The TSF shall resist ***[assignment: heatsink removal and tamper events]*** to the ***[assignment: LS2 HSM Adapter]*** by responding automatically such that the SFRs are always enforced.

FPT_STM.1: Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application Note

The TOE provides timestamps suitable for supporting the time in the audit records for FAU_GEN.1.

FPT_STM.2: Time Source

FPT_STM.2.1 The TSF shall allow the ***[assignment: authorized master crypto officer (MCO) user]*** *to* ***[selection: set the time].***

Application Note

The TOE includes a Real-Time Clock (RTC) which is used to timestamp various operations. The RTC must be configured and maintained properly to ensure accurate timestamps.

FPT_TST.1: TSF Self Testing

- **FPT_TST.1.1** The TSF shall run a suite of the following self-tests ***[selection: during initial start-up, periodically during normal operation, at the request of a user, at the conditions [assignment: Conditional tests are performed every time an operation that requires the listed algorithms is done]]*** to demonstrate the correct operation of ***[selection: the TSF]: [assignment: [At initial start-up (or power-on):***
  - ***Software/firmware integrity test***
  - ***Cryptographic algorithm tests***
  - ***Random number generator tests***
  - ***SP 800-90B startup health tests (RCT and APT)***
  - ***Firmware load test (RSA Signature Verification) – RSA 2048-SHA-512/ECC-P521-SHA512].***
- ***Periodic/at the request of a user***
  - ***Cryptographic algorithm tests***
  - ***Random number generator tests***
  - ***DRBG, SP800-90A Health Tests***
  - ***SP 800-90B (HW RNG) continuous health tests (RCT and APT)***

- ***Conditional tests are performed every time an operation that requires the listed algorithms is done***

  - ***ECDSA Pairwise Consistency Test, for every time EC Key pair is generated.***
  - ***RSA Pairwise Consistency Test, for every time RSA key pair is generated.***
  - ***DRBG, SP800-90A Health Tests, when TRNG is accessed.***
  - ***SP 800-90B (HW RNG) continuous health tests (RCT and APT), When TRNG is accessed.***
  - ***SP 800-56A rev3 required assurances (All SP 800-56Ar3 implementations).***

- ***Firmware load test (RSA Signature Verification) – RSA 2048-SHA-512/ECC-P521-SHA512] while upgrading the FW.***

Application Note

Start-up tests are also executed periodically in a configurable period, set to every 24 hours by default. They can also be requested by authorized users.

Cryptographic tests specified and firmware load test are executed automatically whenever a cryptographic operation using these algorithms is executed.

FPT_TST.1.2

The TSF shall provide authorized users with the capability to verify the integrity of ***[selection: [assignment: secret key value and attributes]***.

FPT_TST.1.3

The TSF shall provide authorized users with the capability to verify the integrity of ***[selection: [assignment: Sotware/Firmware image].***

FMT: SECURITY MANAGEMENT

FMT_MSA.1/GenKeys: Management of security attributes – (General Keys)

FMT_MSA.1.1/GenKeys The TSF shall enforce the *Key Usage SFP* to restrict the ability to *modify* the security attributes ***[assignment: Key attribute specified in Table 15]*** to ***[assignment: Modification conditions specified in Table 15]***.

Application Note

Key attribute	Correspondence with PP key attribute (MSA.1)	General keys
Reference ID (not a key attribute, automatically generated with the key for uniquely identify it and internally handled by the TSF)	Key ID	Cannot be modified
OBJ_ATTR_PRIVATE, OBJ_ATTR_KEY_TYPE	Key type	Cannot be modified
(Not a key attribute) The TSF internally handles, for each partition, a list matching users unique IDs with Keys' Reference ID. This, together with the user credentials, make up the required "Authorization Data".	Authorization Data	Case of a PCU: the owner can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data, or by PCO. Case of the MCO: the owner can be modified by Key Owner only when modification operation includes

Key attribute	Correspondence with PP key attribute (MSA.1)	General keys
		successful validation of current (pre-modification) authorization data.
OBJ_ATTR_ENCRYPT, OBJ_ATTR_DECRYPT, OBJ_ATTR_WRAP, OBJ_ATTR_UNWRAP, OBJ_ATTR_SIGN, OBJ_ATTR_VERIFY, OBJ_ATTR_DERIVE	Key Usage	If OBJ_ATTR_MODIFIABLE = true, the attributes can be modified by the owner of the key.
OBJ_ATTR_EXTRACTABLE, OBJ_ATTR_NEVER_EXTRACTABLE	Export Flag	If OBJ_ATTR_MODIFIABLE = true, OBJ_ATTR_EXTRACTABLE can be modified by the owner of the key. If OBJ_ATTR_NEVER_EXTRACTABLE = true, it can never be modified.
OBJ_ATTR_EKCV	Integrity Protection Data	Cannot be modified
OBJ_ATTR_ALWAYS_AUTHENTICATE	NA	If TRUE, the user must authenticate for each use (sign or decrypt) with the key. If OBJ_ATTR_MODIFIABLE = true, the attribute can be modified by the owner of the key.
OBJ_ATTR_CC_KEY_HASH	Integrity check for key and key attributes	If any of the attributes are updated, this attribute will be recalculated and stored.

Table 15 General keys attributes modification supporting table

FMT_MSA.1/AKeys: Management of security attributes – (Assigned keys)

FMT_MSA.1.1/AKeys The TSF shall enforce the *Key Usage SFP* to restrict the ability to *modify* the security attributes ***[assignment: Key attribute specified in Table 16]*** to ***[assignment: Modification conditions specified in Table 16]***.

Application Note

Key attribute	Correspondence with PP key attribute (MSA.1)split	Assigned keys
Reference ID (not a key attribute, automatically generated with the key for uniquely identify it and internally handled by the TSF)	Key ID	Cannot be modified

Key attribute	Correspondence with PP key attribute (MSA.1)split	Assigned keys
OBJ_ATTR_PRIVATE, OBJ_ATTR_KEY_TYPE	Key type	Cannot be modified
(Not a key attribute) The TSF internally handles, for each partition, a list matching users unique IDs with Keys' Reference ID. This, together with the user credentials, make up the required "Authorization Data".	Authorization Data	Can be modified by Key Owner only when modification operation includes successful validation of current (pre-modification) authorization data.
OBJ_ATTR_ENCRYPT, OBJ_ATTR_DECRYPT, OBJ_ATTR_WRAP, OBJ_ATTR_UNWRAP, OBJ_ATTR_SIGN, OBJ_ATTR_VERIFY, OBJ_ATTR_DERIVE	Key Usage	Cannot be modified
OBJ_ATTR_NEVER_EXTRACTABLE	Export Flag	Cannot be modified
The combination of OBJ_ATTR_MODIFIABLE == false, OBJ_ATTR_NEVER_EXTRACTABLE == true and OBJ_ATTR_LOCAL== true provide the functionality of the "assigned flag"	Assigned Flag	Cannot be modified
OBJ_ATTR_EKCV	Integrity Protection Data	Cannot be modified
OBJ_ATTR_ALWAYS_AUTHENTICATE	NA	If TRUE, the user must authenticate for each use (sign or decrypt) with the key. If OBJ_ATTR_MODIFIABLE = true, the attribute can be modified by the owner of the key.
OBJ_ATTR_CC_KEY_HASH	Integrity	Cannot be modified.

Table 16 Assigned keys attributes modification supporting table

FMT_MSA.3/Keys: Static attribute initialisation

FMT_MSA.3.1/Keys The TSF shall enforce the *Key Usage SFP* to provide ***[selection: restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Keys The TSF shall allow the ***[assignment: Partition Crypto User (PCU)]*** to specify alternative initial values to override the default values when an object or information is created.

Application Note

The PCU can specify alternative initial values only in some of the key attributes, following this table:

Key Attribute (MSA.1)	Assigned Key	Other Key
Reference ID	Initialized by generation process.	Initialized by generation process.
OBJ_ATTR_PRIVATE	Initialized by generation process.	Initialized by generation process.
OBJ_ATTR_KEY_TYPE	Initialized by generation process.	Initialized by generation process.
Owner of the key	Initialized by generation process and assigned to the user that generated the key.	Initialized by generation process and assigned to the user that generated the key.
OBJ_ATTR_ENCRYPT, OBJ_ATTR_DECRYPT, OBJ_ATTR_WRAP, OBJ_ATTR_UNWRAP, OBJ_ATTR_SIGN, OBJ_ATTR_VERIFY, OBJ_ATTR_DERIVE	When a key is generated the KeyUsage attributes are specified as parameters for the key generation command.	When a key is generated the KeyUsage attributes are specified as parameters for the key generation command. This attribute can be modified after key generation if the attribute OBJ_ATTR_MODIFIABLE == true
OBJ_ATTR_EXTRACTABLE, OBJ_ATTR_NEVER_EXTRACTABLE	False (i.e.; no export allowed)	Initialised by generation process (default: false, i.e. no export allowed)
OBJ_ATTR_EKCV	Initialised automatically by TSF	Initialised automatically by TSF
OBJ_ATTR_ALWAYS_AUTHENTICATE	Initialized by generation process.	If TRUE, the user must authenticate for each use (sign or decrypt) with the key. If OBJ_ATTR_MODIFIABLE =

Key Attribute (MSA.1)	Assigned Key	Other Key
		true, the attribute can be modified by the owner of the key.
OBJ_ATTR_CC_KEY_HASH	Computed in generation process	Cannot be modified

Table 17 Key attributes initialization supporting table

FMT_MTD.1/Unblock: Management of TSF data

FMT_MTD.1.1/Unblock The TSF shall restrict the ability to *unblock* the ***[assignment: blocked TSF data specified in Table 18]*** to ***[assignment: unblocker role specified in Table 18]***.

Application Note

Blocked TSF data	Unblocker role
Master Crypto Officer account	None.
Pre-Crypto Officer account	N/A
Partition Crypto Officer account	Master Crypto Officer
Partition Crypto User account	Partition Crypto Officer
Appliance User	Partition Crypto Officer

Table 18 Unblocking of TSF data supporting table

FMT_MTD.1/AuditLog: Management of TSF data

FMT_MTD.1.1/AuditLog The TSF shall restrict the ability to *control export and deletion of* the *audit log records* to *Administrator role*.

Application note:

The TOE do not include the *Administrator* role in the literal sense, but their expected functions are available to the other TOE roles according to the definition of FMT_SMR.1.

In the case of this component, the user is Appliance User (AU).

FMT_SMF.1: Specification of Management Functions

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

(1) Unblock of access due to authentication or authorisation failures

(2) Modifying attributes of keys

(3) Export and deletion of the audit data, which can take place only under the control of the Administrator role

(4) ***[selection: backup and restore functions]***

(5) ***[Selection: key import function]***

(6) ***[selection: key export function]***

Application note:

The TOE do not include the *Administrator* role in the literal sense, but their expected functions are available to the other TOE roles according to the definition of FMT_SMR.1

FMT_SMR.1: Security roles

FMT_SMR.1.1 The TSF shall maintain the roles *Administrator,* ***[selection: Local Client Application, External Client Application]****, Key User,* ***[assignment: Master Crypto Officer (MCO), Pre-Crypto Officer (Pre-CO), Partition Crypto Officer (PCO), Partition Crypto User (PCU), Appliance User (AU)]***.

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

Application note:

The TOE do not include the *Administrator* and *Key User* roles in the literal sense, but their expected functions are available to the other TOE roles defined in this component according to the following table:

Protection profile	This ST
Administrator	Master Crypto Officer (MCO) Pre-Crypto Officer (Pre-CO) Partition Crypto Officer (PCO) Appliance User (AU)
Key User	Partition Crypto User (PCU)

Table 19 User role mapping

FAU: SECURITY AUDIT

FAU_GEN.1: Audit data generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the *not specified* level of audit; and

c) Startup of the TOE;

d) Shutdown of the TOE;

e) Cryptographic key generation (FCS_CKM.1);

f) Cryptographic key destruction (FCS_CKM.6);

g) Failure of the random number generator (FCS_RNG.1);

h) Authentication and authorization failure handling (FIA_AFL.1): all unsuccessful authentication or authorization attempts, the reaching of the threshold for the unsuccessful authentication or authorization attempts and the blocking actions taken;

i) All attempts to import or export keys (FDP_IFF.1/KeyBasics);

j) All modifications to attributes of keys (FDP_ACF.1/KeyUsage, FMT_MSA.1/GenKeys and FMT_MSA.1/Akeys);

k) Backup and restore (FDP_ACF.1/Backup): use of any backup function, use of any restore function, unsuccessful restore because of detection of modification of the backup data;

l) Integrity errors detected for keys (FDP_SDI.2);

m) Failures to establish secure channels (FTP_TRP.1/Local, FTP_TRP.1/External);

n) Self-test completion (FPT_TST.1);

o) Failures detected by the TOE (FPT_FLS.1);

p) All administrative actions (FMT_SMF.1, FMT_MSA.1 (all iterations), FMT_MSA.3/Keys);

q) Unblocking of access (FMT_MTD.1/Unblock);

r) Modifications to audit parameters (affecting the content of the audit log) (FAU_GEN.1)

s) [assignment:

1. FCS_CKM.5 –

minimal: Success and failure of the activity.

2. FCS_COP.1 –

minimal: Success and failure, and the type of cryptographic operation.

basic: Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.

3. FCS_RBG.1 –

minimal - Failure of the randomization process, failure to initialize or reseed (as supported by the technology)

4. FDP_IFF.1/Partitions –

minimal: Decisions to permit requested information flows.

basic: All decisions on requests for information flow.

5. FIA_UAU.1 –

minimal: Unsuccessful use of the authentication mechanism.

basic: All use of the authentication mechanism.

6. FIA_UAU.6 –

minimal: Failure of re-authentication.

basic: All re-authentication attempts.

7. FIA_UID.1 –

minimal: Unsuccessful use of the user identification mechanism, including the user identity provided.

basic: All use of the user identification mechanism, including the user identity provided.

8. FMT_SMR.1 -

minimal: modifications to the group of users that are part of a role.

9. FPT_STM.1 -

minimal: Changes to the time.

detailed: Providing a timestamp.

10. FPT_STM.2 -

minimal: Discontinuous changes to the time.

11. FMT_MTD.1/AuditLog –

basic: All modifications to the values of TSF data.].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each auditable event type, based on the auditable event definitions of the functional components included in the PP, PP-Module, functional package or ST ***[assignment: none]***.

FAU_GEN.2: User identity association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.3: Guarantees of audit data availability

FAU_STG.3.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.3.2 The TSF shall be able to ***[selection: prevent]*** unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.3.3 The TSF shall ensure that ***[assignment: all]*** stored audit records will be maintained when the following conditions occur: ***[selection: audit storage exhaustion]***.

5.2 SECURITY ASSURANCE REQUIREMENTS

The development and the evaluation of the TOE shall be done in accordance to the following security assurance requirements: **EAL4 + AVA_VAN.5, ALC_FLR.3**

The following table shows the assurance requirements by reference the individual components in **[CC2022]**.

Assurance Class	Assurance Components
ASE: Security Target evaluation	ASE_CCL.1: Conformance claims ASE_ECD.1: Extended components definition ASE_INT.1: ST introduction ASE_TSS.1: TOE summary specification ASE_OBJ.2: Security objectives ASE_REQ.2: Derived security requirements ASE_SPD.1: Security problem definition
ALC: Life-cycle support	ALC_DEL.1: Delivery procedures ALC_LCD.1: Developer defined life-cycle model ALC_TAT.1: Well-defined development tools ALC_CMC.4: Production support, acceptance procedures and automation ALC_CMS.4: Problem tracking CM coverage ALC_DVS.1: Identification of security measures **ALC_FLR.3: Systematic flaw remediation**
ADV: Development	ADV_ARC.1: Security architecture description ADV_FSP.4: Complete functional specification ADV_IMP.1: Implementation representation of the TSF ADV_TDS.3: Basic modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance AGD_PRE.1: Preparative procedures
ATE: Tests	ATE_COV.2: Analysis of coverage ATE_DPT.1: Testing: basic design ATE_FUN.1: Functional testing ATE_IND.2: Independent testing – sample
AVA: Vulnerability assessment	**AVA_VAN.5: Advanced methodical vulnerability analysis**

Table 20 Security Assurance Requirements

REFINEMENTS OF SECURITY ASSURANCE REQUIREMENTS

The following refinements are made to selected assurance requirements in Table 20:

ADV_ARC.1 Security architecture description

Refinement:

The following specific topics must be addressed as part of ADV_ARC.1 for this Protection Profile. It is acceptable for references to deliverables supplied for other assurance families, such as ADV_FSP, to be used to meet these requirements, provided that the relationship of the relevant interface specifications to the concepts in the Protection Profile is clear. Note that in some cases, the requirement for description of these particular aspects under ADV_ARC is intended to make clear any differences between the full capabilities of the product and the scope of the Security Target.

1. In general, cryptographic modules will make use of 'support keys' as part of their implementation of protection mechanisms, where these keys are generally not held on behalf of specific users or client applications, but are used by the TOE to carry out its normal operations and as part of the implementation mechanism other SFRs and to protect the TSF itself. These support keys may be used for a variety of purposes (including aspects such as authentication, authorisation, secure channels, security of external storage, or internal data protection), For the purposes of this PP, support keys used by the TOE are treated as TSF data, and require a specific security rationale to be included as part of the ADV_ARC.1 deliverables. This rationale must include a description of the key architecture, identifying all support keys used by the TOE (at least in its evaluated configuration), their method of generation and storage, their purpose in TOE operation, and the ways in which they are protected so as to support the requirements of FDP_IFF.1/KeyBasics and FDP_ACF.1/KeyUsage (noting that the mechanisms used for support keys may differ from those used for user keys). Examples would be keys used for wrapping user keys in order to allow secure storage of the user keys, keys used to implement secure channels, and keys used to protect backups. The description must demonstrate that sufficient entropy has been used in the generation of each support key, and the source of that entropy. The rationale must demonstrate that these support keys cannot be exported/imported in a way that threatens the secure operation of the TOE. The evaluator shall include the description of the support keys in their analysis of the protection of user data (e.g. to confirm that it does not introduce vulnerabilities in the implementation of the SFRs).
2. If updates to the TOE software or firmware are supported then the ADV_ARC.1 deliverables must describe how the TOE is protected against unauthorised updates, by using digital signatures. This shall be confirmed by evaluator testing (if updates are supported) to confirm that updates with invalid signatures are rejected without being executed. The digital signature algorithms used to protect updates shall be included in the scope of FCS_COP.1 signature SFR(s).
3. The ADV_ARC.1 deliverables must in particular describe
    a. Any use that the TOE makes of an audit server
    b. The locations used for any externally stored keys and the structure and format of the externally stored keys including the cryptographic structures that protect the keys in their externally stored form, and that bind them to their attributes (support keys are separately addressed by the description required in item 1 above)
    c. All key import and/or export functions and the secure channels that they use
    d. The secure channels supported by the TOE and the authentication mechanisms that they use (cf. FTP_TRP.1/Local & FTP_TRP.1/External)

e. All local and external interfaces used for communications with users, client applications, audit data, and stored TOE data (cf. Figure 1)
f. The specific key attributes supported, their method of representation (e.g. the relevant data structures and permitted values) and the method by which they are bound to the corresponding key value (cf. FMT_MSA.1). This also includes identifying the types of keys (if any) that support re-authorisation conditions described in FIA_UAU.6/KeyAuth
g. The user types and roles supported, the interfaces by which they interact with the TOE (e.g. a local administrator console or an externally available API), the authentication methods used (cf. FIA_UAU.1 and Application Note 17), and any privileges available to the user type/role
h. All of the cryptographic functions provided (cf. section 1.3.1.1) and whether any nonendorsed cryptographic algorithms and/or cryptographic functions are available (cf. FCS_COP.1 and section 1.3.1.3)
i. The authorisation methods used for keys (cf. FIA_UAU.6/KeyAuth & FDP_ACC.1/KeyUsage)
j. Description of the way in which the TOE ensures that it only holds authorisation data for the minimum time possible before deallocating it according to FDP_RIP.1
k. If the TOE provides backup operations then the ADV_ARC deliverables shall describe the use of support keys by the backup and restore processes (cf. FDP_ACF.1/Backup), and in particular shall describe the ways in which confidentiality and integrity of the backup are provided, and the way in which the TOE rejects an attempt to carry out a restore process using backup data that has been modified
l. Any mechanisms that the TOE uses to support dual person control (cf. FDP_ACF.1/Backup).

AGD_OPE.1 Operational user guidance

Refinement:

The following specific topics must be addressed as part of the Operational Guidance for the TOE:

1. The specific ways in which the TOE needs to be configured and used in order to provide qualified electronic signatures and qualified electronic seals that meet the requirements of [Regulation]. This includes ways in which the TOE can ensure that the signatory can, with high level of confidence, have sole control over the use of the secret key that acts as his/her signature creation data. Thus, for example, it may be necessary for client applications to use TOE interfaces according to certain guidance in order to correctly implement the requirements on attributes of keys as described in this PP. It may be necessary for the TOE to define ways in which secret keys to be used for signing purposes can be created in a way that does not allow subsequent modification of some or all of their attributes, e.g. by an administrator, before they are assigned to the signatory (cf. FMT_MSA.1/AKeys). The intention of this aspect of the operational user guidance documentation is to identify the configuration and secure use required for a particular TOE, and how it is necessary to connect this with other aspects such as procedural controls and client applications in the operational

environment. The evaluators shall test the identified ways of using the TOE for qualified electronic signatures and qualified electronic seals to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys produced by following the Operational Guidance do indeed meet the requirements of requirements of [Regulation, Annex II & Annex III] for qualified electronic signatures and qualified electronic seals.

2. The use of trusted channels (cf. FTP_TRP.1/Local & FTP_TRP.1/External).
3. The available key attributes, their possible values, and the meaning of each of these values (cf. FMT_MSA.1/GenKeys and FMT_MSA.1/AKeys, including their use to constrain the period and number of uses that are enabled by authorisation of a key (cf. FIA_UAU.6/KeyAuth and Application Note 19).
4. Identification of any non-endorsed cryptographic algorithms and/or cryptographic functions that are available (cf. FCS_COP.1 and section 1.3.1.3).
5. Identification of any other cryptographic algorithms and operations that are not included in the scope of the Security Target.
6. Possible errors from the self-test process and the actions that should be taken in response to each (cf. FPT_TST.1 & Application Note 32).
7. Specific failures detected by the TOE (cf. FPT_FLS.1 & Application Note 35).
8. Audit functions and their configuration (including specification of the available audit records), along with any other actions that are associated with audit functions (e.g. archiving or viewing audit records, or use of an external audit server) (cf. FAU_GEN.1 & Application Note 42, FAU_STG.3 & Application Note 43, FMT_MTD.1/AuditLog & Application Note 39).
9. Any configuration and operation requirements for dual-control operations (cf. FDP_ACF.1/Backup).
10. If backup is provided by the TOE (cf. FDP_ACF.1/Backup), then the Operational Guidance shall describe the backup and restore functions, and the administrator roles that are required to carry them out.
11. If key import is provided by the TOE, then the Operational Guidance shall describe how attributes are defined for any imported keys (cf. FMT_MSA.3/Keys). The evaluators shall test the import process to demonstrate that the description in the Operational Guidance is suitably complete, and that the keys imported have attributes appropriately defined. Similarly, if key export is provided by the TOE then the Operational Guidance shall describe whether attributes are exported with keys (and if so, then how the attributes are represented and associated with the exported key), and the evaluators shall test the export process to demonstrate that the description in the Operational Guidance is suitably complete, and that the handling of attributes is as described.

ATE_IND.2 Independent testing – sample

Refinement:

The following specific topics must be addressed as part of the independent testing of the TOE:

1. The evaluator shall execute the electronic signature and electronic seal operations provided by the TOE and shall confirm that the signatures and seals returned by the TOE correspond to the correct DTBS.

2. If software and/or firmware updates are supported by the TOE then the evaluator shall carry out tests to ensure that only updates with valid digital signatures can be installed on the TOE.

AVA_VAN.5 Advanced methodical vulnerability analysis

Refinement:

Regarding the protection of the TOE against physical attacks: because of the requirement for a physically secure environment with regular inspections (cf. OE.Env), the level of protection (and hence resistance to attack potential) that is required by the implementation of FPT_PHP.1 and FPT_PHP.3 for this TOE is equivalent to the level of assessment in section 7.7.2 Physical security general requirements and section 7.7.3 Physical security requirements for each physical security embodiment in ISO/IEC 19790:2012 for Security Level 3.

5.3 SECURITY REQUIREMENTS RATIONALE

NECESSITY AND SUFFICIENCY ANALYSIS

SFR / TOE Security Objective	OT.Plain	OT.Algor	OT.KeyI	OT.Auth	OT.KeyU	OT.KeyU	OT.Data	OT.Data	OT.Impo	OT.Back	OT.RNG	OT.Tam	OT.Failu	OT.Audi	OT.P_AC	OT.P_M
FCS_CKM.1		X														
FCS_CKM.6	X															
FCS_COP.1		X														
FIA_UID.1				X												
FIA_AFL.1				X												
FPT_TST.1													X			
FIA_UAU.6/KeyAuth				X		X										
FDP_IFF.1/KeyBasics	X		X		X				X							
FDP_ACF.1/KeyUsage					X	X										
FDP_IFC.1/KeyBasics	X				X				X							

SFR / TOE Security Objective	OT.Plain	OT.Algor	OT.KeyI	OT.Auth	OT.KeyU	OT.KeyU	OT.Data	OT.Data	OT.Impo	OT.Back	OT.RNG	OT.Tam	OT.Failu	OT.Audi	OT.P_AC	OT.P_M
FDP_ACC.1/Key Usage					X	X										
FDP_ACC.1/Backup										X						
FDP_ACF.1/Backup										X						
FDP_IFC.1/Partitions															X	X
FDP_IFF.1/ Partitions															X	X
FCS_RNG.1											X					
FCS_RBG.1											X					
FCS_RBG.3											X					
FCS_RBG.6											X					
FCS_CKM.5		X														
FIA_UAU.1				X												
FDP_SDI.2			X													
FDP_RIP.1	X				X											
FTP_TRP.1/Local			X	X			X	X	X							
FTP_TRP.1/External			X	X			X	X	X							
FPT_PHP.1												X				
FPT_PHP.3												X				
FPT_FLS.1													X			

SFR / TOE Security Objective	OT.Plain	OT.Algor	OT.KeyI	OT.Auth	OT.KeyU	OT.KeyU	OT.Data	OT.Data	OT.Impo	OT.Back	OT.RNG	OT.Tam	OT.Failu	OT.Audi	OT.P_AC	OT.P_M
FPT_STM.1														X		
FPT_STM.2														X		
FMT_SMR.1				X										X		
FMT_SMF.1				X										X		
FMT_MTD.1/Unblock				X												
FMT_MTD.1/AuditLog														X		
FMT_MSA.1/GenKeys					X											
FMT_MSA.1/Akeys					X											
FMT_MSA.3/Keys					X											
FAU_GEN.1														X		
FAU_GEN.2														X		
FAU_STG.3														X		

Table 21 SFRs / TOE Security Objectives coverage

SECURITY REQUIREMENT SUFFICIENCY

OT.PlainKeyConf: OT.PlainKeyConf is addressed by the requirements in the Key Basics SFP defined in **FDP_IFC.1/KeyBasics** and **FDP_IFF.1/KeyBasics** (especially **FDP_IFF.1.5/KeyBasics**). Secure destruction of keys according to **FCS_CKM.6** protects the key value at the end of its lifetime. **FDP_RIP.1** protects secret keys from being accessed after they have been deallocated.

OT.Algorithms: OT.Algorithms is addressed by the need to use endorsed standards for **FCS_COP.1** and the use of an appropriate random number generator in **FCS_CKM.1**. Is also addressed by the key derivation mechanisms endorsed in FCS_CKM.5.

OT.KeyIntegrity: OT.KeyIntegrity is addressed primarily by **FDP_SDI.2** which requires integrity protection of keys and their attributes by the TOE. **FDP_IFF.1/KeyBasics** requires that any importing or exporting of keys requires the use of secure channels and integrity protection (cf. the requirement for an integrity-protected channel as part of **FTP_TRP.1/Local** and **FTP_TRP.1/External**, which is linked to the Key Basics SFP under **FDP_IFF.1/KeyBasics**).

OT.Auth: OT.Auth is addressed by **FIA_UID.1**, **FIA_UAU.1** and **FIA_AFL.1** for administrator authentication (with **FMT_MTD.1/Unblock** and its dependencies on **FMT_SMR.1** and **FMT_SMF.1** ensuring that appropriate roles and unblocking for authorisation and authentication failures are also provided). Authorisation for external client applications is provided by the requirements for authentication of endpoints in **FTP_TRP.1/Local** and **FTP_TRP.1/External**. Authorisation for the use of secret keys is addressed by **FIA_UAU.6/KeyAuth**.

OT.KeyUseConstraint: OT.KeyUseConstraint is addressed by the requirements for well-defined (and securely initialised) key attributes in **FMT_MSA.1/GenKeys**, **FMT_MSA.1/Akeys**, and **FMT_MSA.3/Keys**, and the application of the attributes to operate constraints on the use of keys in **FDP_IFC.1/KeyBasics**, **FDP_IFF.1/KeyBasics**, **FDP_ACC.1/KeyUsage** and **FDP_ACF.1/KeyUsage**. **FDP_RIP.1** protects authorisation data (which enables a key to be used) from being accessed after it has been deallocated.

OT.KeyUseScope: OT.KeyUseScope is addressed by the Key Usage SFP in **FDP_ACC.1/KeyUsage** and **FDP_ACF.1/KeyUsage** and by the re-authorisation conditions for use of a secret key specified in **FIA_UAU.6/KeyAuth**.

OT.DataConf: OT.DataConf is addressed by the authentication and confidentiality requirements for secure channels in **FTP_TRP.1/Local** and **FTP_TRP.1/External**.

OT.DataMod: OT.DataMod is addressed by the authentication and integrity requirements for secure channels in **FTP_TRP.1/Local** and **FTP_TRP.1/External**.

OT.ImportExport: OT.ImportExport is addressed by the requirements for the use of secure import/export through a secure channel and restrictions on how keys are imported and exported to protect confidentiality and integrity in the Key Basics SFP in **FDP_IFC.1/KeyBasics** and **FDP_IFF.1/KeyBasics**, and by the requirements on the secure channels themselves in **FTP_TRP.1/Local** and **FTP_TRP.1/External**.

OT.Backup: OT.Backup separates out the requirements for any backup and restore properties that the TOE may provide and is addressed directly by the Backup SFP in **FDP_ACC.1/Backup** and **FDP_ACF.1/Backup**.

OT.RNG: OT.RNG is addressed by the requirement in **FCS_RBG.1 and FCS_RNG.1** for a random number generator of an appropriate type, which meets appropriate randomness metrics. Internal seeding is addressed in FCS_RBG.3, and random bit generator as a service is addressed in FCS_RBG.6.

OT.TamperDetect: OT.TamperDetect is addressed by the requirement for passive tamper detection in **FPT_PHP.1** and the tamper response mechanisms in **FPT_PHP.3**.

OT.FailureDetect: OT.FailureDetect is addressed by the self-test requirements of **FPT_TST.1** and secure failure requirements of **FPT_FLS.1**.

OT.Audit: OT.Audit is addressed in terms of basic creation of audit records by the requirements for audit record generation in **FAU_GEN.1** and **FAU_GEN.2** and provision of timestamps for use in audit records in **FPT_STM.1**. Provision of configuring the Real-Time-Clock (RTC) which is used in timestamp various operations for use in audit records in FPT_STM.2. Protection of the audit trail is ensured by **FAU_STG.3**, **FMT_MTD.1/AuditLog** and **FMT_SMF.1**. Support for the Administrator role that controls export and deletion of audit records from the TOE is required by **FMT_SMR.1**.

OT.P_ACCESSIBILITY is addressed by **FDP_IFC.1/Partitions** and **FDP_IFF.1/Partitions** which guarantee that the partitions will only have access to the information contained in them.

OT.P_MP_FUNCTIONALITY is addressed by **FDP_IFC.1/Partitions** and **FDP_IFF.1/Partitions** which guarantee that User Partitions will not be able to make use of the functionalities reserved for the Master Partition.

SFR DEPENDENCY RATIONALE

Table of SFR dependencies

The following table lists the dependencies for each security functional requirement, indicating how they have been satisfied. The abbreviation "h.a." indicates that the dependency has been satisfied by a SFR that is hierarchically above the required one.

SFR	Required	Fulfilled	Missing
FCS_CKM.1	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1 and FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6	FCS_COP.1 FCS_RBG.1 FCS_CKM.6	None
FCS_CKM.5	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.6	FCS_COP.1 FCS_CKM.6	None
FCS_CKM.6	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	FCS_CKM.1	None
FCS_COP.1/Symmetric	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6	FCS_CKM.1 FCS_CKM.6	None
FCS_COP.1/MAC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.6	FCS_CKM.1 FCS_CKM.6	None
FCS_COP.1/Asymmetric	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.6	FCS_CKM.1 FCS_CKM.6	None
FCS_COP.1/Digest	FCS_CKM.6	FCS_CKM.6	None

SFR	Required	Fulfilled	Missing
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1	
FCS_RNG.1	None	None	None
FCS_RBG.1	[FCS_RBG.2 or FCS_RBG.3] FPT_FLS.1 FPT_TST.1	FCS_RBG.3 FPT_FLS.1 FPT_TST.1	None
FCS_RBG.3	FCS_RBG.1	FCS_RBG.1	None
FCS_RBG.6	FCS_RBG.1	FCS_RBG.1	None
FIA_UID.1	None	None	None
FIA_UAU.1	FIA_UID.1	FIA_UID.1	None
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1	None
FIA_UAU.6/KeyAuth	None	None	None
FDP_IFC.1/KeyBasics	FDP_IFF.1	FDP_IFF.1/KeyBasics	None
FDP_IFF.1/KeyBasics	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/KeyBasics FMT_MSA.3/Keys	None
FDP_ACC.1/KeyUsage	FDP_ACF.1	FDP_ACF.1/KeyUsage	None
FDP_ACF.1/KeyUsage	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/KeyUsage FMT_MSA.3/Keys	None
FDP_ACC.1/Backup	FDP_ACF.1	FDP_ACF.1/Backup	None
FDP_ACF.1/Backup	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Backup	FMT_MSA.3 (see justification below)
FDP_SDI.2	None	None	None
FDP_RIP.1	None	None	None
FTP_TRP.1/Local	None	None	None
FTP_TRP.1/External	None	None	None
FPT_STM.1	None	None	None
FPT_TST.1	None	None	None

SFR	Required	Fulfilled	Missing
FPT_FLS.1	None	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.1	None
FMT_SMF.1	None	None	None
FMT_MTD.1/Unblock	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1	None
FMT_MTD.1/AuditLog	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1	None
FMT_MSA.1/GenKeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1	None
FMT_MSA.1/Akeys	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1/KeyUsage FDP_IFC.1/KeyBasics FMT_SMR.1 FMT_SMF.1	None
FMT_MSA.3/Keys	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/GenKeys FMT_MSA.1/Akeys FMT_SMR.1	None
FAU_GEN.1	FPT_STM.1	FPT_STM.1	None
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.1	None
FAU_STG.3	FAU_GEN.1	FAU_GEN.1	None
FPT_PHP.1	None	None	None
FPT_PHP.3	None	None	None
FDP_IFC.1/Partitions	FDP_IFF.1	FDP_IFF.1/Partitions	None
FDP_IFF.1/ Partitions	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1/Partitions	FMT_MSA.3
FPT_STM.2	FPT_STM.1 FMT_SMR.1	FPT_STM.1 FMT_SMR.1	None

Table 22 SFR Dependencies

Justifications

- The dependency of FDP_ACF.1/Backup on FMT_MSA.3 is not relevant in this case since the attribute used in FDP_ACF.1/Backup is determined by the ability of the user to be authenticated as an administrator according to FIA_UAU.1/Roles.
- The dependency of FDP_IFF.1/Partitions on FMT_MSA.3 is not applicable to this case since this requirement try to describe the roles defined in FMT_SMR.1 that are able to modify security attributes (as is described in FMT_MSA.1). However, the TOE does not allow modification of the security attributes in any case, since they are defined as part of the architecture and cannot be modified at runtime.

SAR DEPENDENCY RATIONALE

The assurance level for this protection profile is **EAL4** augmented with **ALC_FLR.3** and **AVA_VAN.5**.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It is considered the highest level that could be applied to an existing product line without undue expense and complexity. As such, **EAL4** is appropriate for commercial products that can be applied to moderate to high security functions. The TOE described in this protection profile is just such a product.

Augmentation results from the selection of **ALC_FLR.3** alongside **AVA_VAN.5**. All the dependencies of **AVA_VAN.5** are satisfied by other assurance components in the **EAL4** assurance package. **ALC_FLR.3** has no dependencies.

> The inclusion of **ALC_FLR.3** is in accordance with our established company practices, which have been implemented to fulfill customer requirements for flaw reporting and remediation.Table of SAR dependencies

The following table lists the dependencies for each security assurance requirement, indicating how they have been satisfied. The abbreviation “h.a.” indicates that the dependency has been satisfied by a SAR that is hierarchically above the required one.

SAR	Required	Fulfilled	Missing
ASE_CCL.1	ASE_INT.1 ASE_ECD.1 ASE_REQ.1	ASE_INT.1 ASE_ECD.1 ASE_REQ.2 (h.a.)	None
ASE_ECD.1	None	None	None
ASE_INT.1	None	None	None
ASE_OBJ.2	ASE_SPD.1	ASE_SPD.1	None
ASE_REQ.2	ASE_OBJ.2 ASE_ECD.1	ASE_OBJ.2 ASE_ECD.1	None
ASE_TSS.1	ASE_INT.1 ASE_REQ.1 ADV_FSP.1	ASE_INT.1 ASE_REQ.2 (h.a.) ADV_FSP.4 (h.a.)	None
ALC_CMC.4	ALC_CMS.1 ALC_DVS.1	ALC_CMS.4 (h.a.) ALC_DVS.1	None

SAR	Required	Fulfilled	Missing
	ALC_LCD.1	ALC_LCD.1	
ALC_CMS.4	None	None	None
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3 (h.a.)	None
AGD_OPE.1	ADV_FSP.1	ADV_FSP.4 (h.a.)	None
AGD_PRE.1	None	None	None
ATE_IND.2	ADV_FSP.2 AGD_OPE.1 AGD_PRE.1 ATE_COV.1 ATE_FUN.1	ADV_FSP.4 (h.a.) AGD_OPE.1 AGD_PRE.1 ATE_COV.2 (h.a.) ATE_FUN.1	None
AVA_VAN.5	ADV_ARC.1 ADV_FSP.4 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1	ADV_ARC.1 ADV_FSP.4 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1	None
ASE_SPD.1	None	None	None
ALC_DEL.1	None	None	None
ADV_ARC.1	ADV_FSP.1 ADV_TDS.1	ADV_FSP.4 (h.a.) ADV_TDS.3 (h.a.)	None
ADV_IMP.1	ADV_TDS.3 ALC_TAT.1	ADV_TDS.3 ALC_TAT.1	None
ADV_TDS.3	ADV_FSP.4	ADV_FSP.4	None
ALC_DVS.1	None	None	None
ALC_LCD.1	None	None	None
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1	None
ALC_FLR.3	None	None	None
ATE_COV.2	ADV_FSP.2 ATE_FUN.1	ADV_FSP.4 (h.a.) ATE_FUN.1	None
ATE_DPT.1	ADV_ARC.1 ADV_TDS.2 ATE_FUN.1	ADV_ARC.1 ADV_TDS.3 (h.a.) ATE_FUN.1	None
ATE_FUN.1	ATE_COV.1	ATE_COV.2 (h.a.)	None

Table 23 SAR dependencies

6 TOE SUMMARY SPECIFICATION

This chapter describes how the TOE meets the SFRs described at section 5. The TSF will be described by means of a set of security features implemented by the TOE as introduced in sections 1.3 and 0. This detailed description and analysis of the TSF demonstrates how the defined security features of the TOE work together and support each other.

6.1 CRYPTOGRAPHIC FUNCTIONS

The TOE implements a set of cryptographic functions listed in the tables associated with SFRs FCS_CKM.1, FCS_CKM.5, FCS_CKM.6, FCS_COP.1/Symmetric, FCS_COP.1/MAC, FCS_COP.1/Asymmetric, FCS_COP.1/Digest, FCS_RBG.1. They discuss the entire list of algorithms (Cryptographic key generation and derivation, symmetric/asymmetric encryption and decryption, message authentication, digital signature, and message digest algorithms) that are supported by the module within the present security evaluation.

TOE's cryptography is supported on HW engines and FW implementation. In the case of HW engines, they are provided by the Octeon cryptographic module in the main CPU while firmware implementation of the cryptographic algorithms is based on OpenSSLv1.1.1u.

The TOE also generates random numbers using a hash based DRBG (FCS_RBG.1) seeded by a physical RNG provided by two hardware modules in the Octeon main CPU (FCS_RNG.1). Random numbers are used for the following purposes:

- Internal IV (AES-CCM, TLSv1.2).
- External or applications' request for random data.
- Internal Nonce (Per Application nonce, per login nonce, challenge during CO unlock, backup nonce, Cert-Auth nonce, Salt).
- Symmetric key generation.
- RSA key generation (for choosing the primes).
- EC signature generation (Random k, the per-message secret number).

AES-CTR DRBG(FCS_RBG.1) is used for generating internal nonces (Used to generate random data for challenge required during unlocking the CO an to generate an exchange key used during registering the recovery key).

The TSF shall use entropy source as TRNG for obtaining entropy for initialization and reseeding (FCS_RBG.3).

The TOE shall be able to seed the DRBG using a hardware-based noise source (FCS_RBG.1.1) and a minimum of 1584 Bytes (12672 bits) entropy.

The TOE also has a service to entities outside of the TOE with an interface to generate required random output (FCS_RBG.6).

All of them are algorithms implemented to support the TSP functionality of the TOE. However, some of the algorithms listed support the security functionality of the TOE:

Secure channels: Its security is supported by the implementation of TLSv1.2, whose cryptographic algorithms are contained in the only cipher suite supported by the TOE's TLS client: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

Key encryption: When the TOE imports or exports keys, they are always in an encrypted form. The TOE is supported by the following cryptographic algorithms for key encryption/decryption. AES KW, AES KWP. (FCS_COP.1/Symmetric)

Backups: TOE backups are encrypted using AES-256-CBC and AES-256-KWP (refer to section 6.3 for additional information).

Updates: The TOE uses RSA PKCS#1 v1.5 or ECDSA algorithm to sign the firmware update files.

Keys integrity: The TOE uses OBJ_ATTR_EKCV attribute of the keys to store a fingerprint of them. For symmetric keys, OBJ_ATTR_EKCV stores a SHA256 of the key generated. In the case of symmetric keys, OBJ_ATTR_EKCV stores a HKDF of the key data whose integrity is granted by the primitive of the algorithm HMAC-SHA256 (FCS_COP.1/Digest).

6.2 KEY MANAGEMENT

The TOE handles two types of keys in accordance with the PP: secret keys and support keys.

Secret keys

Secret keys are the keys on which most of the SFRs in this Security Target are oriented.

Secret keys are generated by TOE users through client applications and they can be used for different purposes as the user wants to use them controlled by API/SDK commands. These keys can be generated, managed and deleted by the TOE users. These keys may support cryptographic algorithms as described at section 6.1.

User keys have attributes stored along with the keys which store information of the keys. The TOE manages key owners through an internal table linking keys to users. The entries in this table can only be modified by the users referenced in this entry (owners of the key), requiring prior password-based login.

Support keys

The TOE makes use of support keys as part of its implementation of protection mechanisms. These keys are not held on behalf of specific users. They are used by the TOE to carry out its normal operations, as part of the implementation mechanism and to protect the TSF itself. These keys are listed at Table 24 Support keys.

Key	Type	Description/Usage
Vendor support keys		
HW_ROT_SB	EC P256 public key	**HW Root of Trust for Secure Boot** Public key, common to all Marvell HSMs to protect bootloader image integrity.

Key	Type	Description/Usage
		Imported during manufacturing.
HW_UEK	AES 256 key	**Hardware Unique Encryption Key** Key used to protect (encrypt) the following vendor-programmed keys: • HID (also encrypted by MCO_RK). • Master_DEK (NOR copy) • VKBK (also encrypted by MCO_RK). This key is generated inside TOE during manufacturing.
HW_EK	ECC P256 private key	**HW Endorsement Key** A unique private key used to endorse (identify) the HSM. Used during the recovery key registration protocol. This key is generated inside TOE during manufacturing.
HID **MAK**	RSA 4096 and EC P521 private keys	**HSM Identity Key (Master Authentication Key)** Each HSM has an RSA and EC key pairs, unique to the HSM. Used to identify that the HSM is developed by Marvell. This key is generated inside TOE during manufacturing using AES-256-KWP with key derived from the HW_UEK. During recovery key registration, this key is double encrypted with the recovery key (MCO_RK).
ROT_HID	RSA 4096 and EC P521 (Two separate files).	**Root of Trust for HSM ID** Public root certificate from Marvell. Each HSM manufactured by Marvell is certified using the Marvell Root Certificate, called Root of Trust for HSM Identity. Marvell has two root certificates: one is RSA and the other is ECC. Every HSM carries these certificates, which are hosted on the Marvell website. Users can read these certificates and cross-check them.

Key	Type	Description/Usage
		This certificate is Imported during TOE manufacturing.
HID_Cert **MAC**	RSA 4096 ECC P521	**HSM Identity Certificate (Master Authentication Certificate)** x509 certificate issued by ROT_HID which is unique to each HSM for both RSA and EC Identity Key. This certificate links to Root of Trust for HSM ID and is unique to each HSM for both RSA and EC Identity Key (MAK/HID). This certificate is Imported during TOE manufacturing.
VKBK	AES 256 key	**Vendor Key Backup Key component** Key common to a unique part or customer used to protect backups. This key is generated inside TOE during manufacturing using AES256-KWP with key derived from the HW_UEK. During recovery key registration, this key is double encrypted with the recovery key (MCO_RK).
FWSK	RSA 2048 key	**Vendor Firmware Signing Key** Key common to all HSMs for verifying the firmware update code before its installation in the TOE. It is part of the FW image and is imported within the firmware.
Licensing cert	RSA 2048	Certificate for licensing check of the HSM. It is part of the FW image and is imported within the firmware.
LS1 Marvell RootCA Cert	RSA 2048	Supports cert-auth feature (used for cloning and smart-card backup with LS1 HSMs). It is part of the FW image and is imported within the firmware.
MCO support keys		
MCO_ROT **AOTA** **AOTAC**	Those listed in FCS_CKM.1	**MCO Root of Trust (Adapter Owner Trust Anchor)** RSA public key certificate common to all HSMs of the customer which is used to identify the HSM owner for STM locking process.

Key	Type	Description/Usage
		This certificate is imported by adapter owner.
MCO_HID **AOAC**	RSA 4096 ECC P521	**MCO-issued HSM Master Partition ID (Adapter Owner Authentication Certificate)** Cross-signed certificate for HID (MAK) linking to MCO Root of Trust (AOTA). This certificate is imported by adapter owner.
MCU_DEK	AES 256 key	**MCU Root Data Encryption Key** AES key used to encrypt (directly or indirectly) all other contents on the HSM. It is never exported out of the HSM. Used to encrypt (directly or indirectly) all other contents on the HSM: • Direct: it directly encrypts directly the MCO_RK and Master_DEK. • Indirect: it indirectly encrypts all keys stored on flash (NOR and eMMC).
MCO_RK	AES 256 key	**MCO Recovery Key** Key used for recovering an HSM from MCO-induced factory reset. Used for recovering an HSM from MCO-induced factory reset (encrypts HID, MKBK and VKBK). This key is imported by adapter owner.
MCO_KBK **AOKBK** **MKBK**	AES 256 key	**MCO KBK Component (Adapter Owner KBK)** Used to derive the VKBK key. This key is imported by adapter owner and can be a common key as a decision of the MCO.
MCO FW Signing Key	Those listed in FCS_CKM.1	**RSA public key certificate/EC Public Key Certificate** MCO FW signing key registration certificate. Possibly common to all HSMs owned by MCO. The MCO FW signing key registration controls downgrades in 2.08 and higher releases. This key is imported by adapter owner.
Master_DEK	AES 256-bit key	**Master Partition Data Encryption Key.**

Key	Type	Description/Usage
		Key used to encrypt the contents of the Master Partition and all PCO keys. The location of this key depends on whether or not the MCO registers a recovery key. This key is never exported out of the HSM. - If a recovery key is not registered, this key is stored in NOR flash. - If a recovery key is registered, this key is stored in the MCU and protected by the MCU_DEK. All subsequent keys (partition-level system keys) are encrypted with this key. This key is generated by HSM during initialization of master partition.
PCO support keys		
PAK **PID**	RSA 2048 or EC key pair	**Partition ID Key or Partition Authentication Key** Key pair unique to a partition. It is used to sign PID_Cert. This key is generated by HSM at the time of partition creation.
PAC **PID_Cert**	RSA 2048 or EC Key	**Partition ID Cert or Partition Authentication Certificate** x509 certificate linking to HSM ID Cert (MAC). Unique to a partition. This certificate is generated by HSM at the time of partition creation.
PCO_ROT **POTA**	Those listed in FCS_CKM.1	**PCO Root of Trust (Partition Owner Trust Anchor)** Public key certificate common to all partitions of a customer. This certificate is imported by partition owner.
PCO_PID **POAC**	RSA 2048	**PCO issued Partition ID (Partition Owner Authentication Cert).** Cross-signed certificate for Partition-ID linking to PCO Root of Trust; same as POAC. This certificate is imported by partition owner

Key	Type	Description/Usage
Part_MEK	AES 256 key	**Partition Master Encryption Key.** Key used to encrypt the contents of a partition. Generated during partition creation. This key is generated by HSM owner
Part_DEK	AES 256 key	**Partition Data Encryption Key.** Key used to encrypt the PID/PAK and PCO KBKs This key is generated during partition creation.
PKBK **PCO_KBK** **POKBK**	AES 256 key	**PCO KBK Component.** Per partition key loaded by the PCO to use in 3-key backup/restore operation. Imported by partition owner
Partition's Masking Key	AES 256 key	Used to import/export CSPs and masked objects. This key is generated during partition initialization or imported during cloning process.

Table 24 Support keys

The TOE provides key management functions as well as internal and external storage. Key management cannot be done without user authentication since all management functions require user authentication.

The keys under control of the TOE (not support keys) can be generated based on FCS_CKM.1/FCS_CKM.5 key generation/derivation algorithms and destructed based on FCS_CKM.6 key destruction mechanisms. The random numbers required for key generation are generated through the internal random number generator (FCS_RNG.1/FCS_RBG.1). The entropy source used for the derivation of these numbers is provided by the Octeon through a non-deterministic random generation method: the OCTEON TX2 HW unit generates random bits from the 8 free-running oscillators from a total of 128 free-running oscillators. The generated random bits are already run through HW-level health tests (APT and RCT) (FPT_TST.1). OCTEON TX2 produces 2.673 bits of minimum entropy in each 16 Bytes (128 bits) of output of its HW RNG.

Keys can also be imported and exported (FDP_IFC.1/Keybasics) through trusted paths (FTP_TRP.1/Local, FTP_TRP.1/Remote) in an encrypted form. When keys are imported, they are marked with the attribute OBJ_ATTR_LOCAL=false, which will prevent them from being used as Assigned Key (FDP_IFC.1/Keybasics, FDP_IFF.1.1/Keybasics, FDP_IFF1.2/Keybasics, FDP_IFF1.3/keybasics, FDP_IFF1.4/keybasics, FDP_IFF1.5/keybasics). Those keys considered as Assigned Keys cannot be exported as they shall be marked with the attribute OBJ_ATTR_EXTRACTABLE=false (FDP_IFF.1.5/KeyBasics).

For managing the keys and control the access to them, the TOE implements an access control policy (Key Usage SFP) for allowing the cryptographic operations to be used (FCS_COP.1/Symmetric, FCS_COP.1/MAC, FCS_COP.1/Asymmetric, FDP_ACC.1/KeyUsage, FDP_ACF.1/KeyUsage) based on security attributes defined when each key is generated (FMT_MSA.3/Keys) and that can be modified depending on the nature of the key (FMT_MSA.1/GenKey, FMT_MSA.1/Akeys). Moreover, the Key Usage SFP controls that only the users specifically authorized for the use of a key(FIA_UAU.6.1/KeyAuth, FDP_IFF1.5/keybasics) can change the key's attributes and use its plaintext value for cryptographic operations.

A secure channel for export of keys in FDP_IFF.1.2/KeyBasics (1) or for import of keys in FDP_IFF.1.2/KeyBasics (3) is one that meets the requirements of FTP_TRP.1/Local or FTP_TRP.1/External.

Unblocking a key as in FDP_IFF.1.2/KeyBasics (6) is intended only to restore the ability of subjects to authorize for access to a key by presenting the correct authorization data.

The encrypted form required for keys imported or exported over a secure channel requires encryption of the key itself, in addition to any encryption provided by the secure channel, it is provided by FDP_IFF1.5/keybasics.

The following table shows the default values of the key attributes(FDP_IFC.1/KeyBasics) when keys are generated to which the TSF enforces restrictive values (FMT_MSA.3/Keys):

Attribute Name	Default Value		
	Public Key	**Private Key**	**Secret Key**
OBJ_ATTR_PRIVATE	TRUE	TRUE	TRUE
OBJ_ATTR_KEY_TYPE	No specific default value, it depends on the key that is being generated.	No specific default value, it depends on the key that is being generated.	No specific default value, it depends on the key that is being generated.
OBJ_ATTR_OWNER	1	1	1
OBJ_ATTR_ENCRYPT	FALSE	FALSE	FALSE
OBJ_ATTR_DECRYPT	FALSE	FALSE	FALSE
OBJ_ATTR_WRAP	FALSE	FALSE	FALSE
OBJ_ATTR_UNWRAP	FALSE	FALSE	FALSE
OBJ_ATTR_SIGN	FALSE	FALSE	TRUE
OBJ_ATTR_VERIFY	FALSE	FALSE	TRUE
OBJ_ATTR_DERIVE	FALSE	FALSE	FALSE
OBJ_ATTR_EXTRACTABLE	TRUE	TRUE	TRUE
OBJ_ATTR_NEVER_EXTRACTABLE	FALSE	FALSE	FALSE

Attribute Name	Default Value		
	Public Key	**Private Key**	**Secret Key**
OBJ_ATTR_EKCV	SHA-256 hash computed on public key.	SHA-256 hash computed on public key.	HKDF, HMAC-SHA256 of the zero block with the generate key.

Table 25 Attributes default values

The use of the keys is controlled by FIA_UAU.6/KeyAuth, granting that authentication is required to use them. Keys are not blocked due to unsuccessful use attempts since their use is authorized once the key's owner successfully performs initial session authentication. The authorization remains valid until an explicit rescinding of previous authorization for access to the secret key.

Additionally, keys are destroyed from volatile memory after its use and CPT microcode resets the core before processing any new request (FCS_CKM.6, FDP_RIP.1).

The integrity of the keys is protected based on key's attribute OBJ_ATTR_EKCV which protects the value of the key and its other security attributes, including when the key is externally stored by the TOE (FDP_SDI.2).

Keys can be backed-up when required by making a partition backup, protected with PKBK (FDP_ACF.1/Backup, FMT_SMF.1).

Key destruction

Keys destruction is based on zeroization (FCS_CKM.6). Support keys are zeroized as described in the events from Table 9 Key destruction support table.

Cryptographic keys can be zeroized as part of a partition deletion, individually or as part of a tamper event:

- Partition deletion: when a partition is deleted, all the sensitive data (keys, users and contexts) is securely deleted by zeroizing them in the RAM. The partition's DEK is destroyed which will cause virtual erasure of the partition's DB in the eMMC (which contains Users and Keys) as it makes the decryption unavailable.
- Single key delete: when a single key delete is performed (by user command to request specific user key destruction), the key is securely erased by zeroizing it from key-cache in the RAM. The key entry is also deleted from the partition's keys DB. In this scenario in which entire partition is not deleted partition's DEK will still be intact and key data overwrite in DB or crypto erasure of the key is not done.
- Tamper event: when a tamper event is raised, support keys (MAK, VKBK, MCO_ROT, MCO_RK, MCO_KBK, MCU_DEK, MCO FW signing Key, Master_DEK) are zeroized to make the dependent user keys and recovery keys inaccessible.

6.3 BACKUP

The backup is made at partition level. The backup of a partition contains the entire configuration of the partition, including the policies configured by the PCO. These same policies will be configured for the restored partition as well (FDP_ACC.1/backup, FDP_ACF.1/backup, FMT_SMF.1).

The backup process consists of encrypting the following components with specified algorithms:

Partition backup content	Encryption algorithm
Configuration (i.e., partition information)	AES-256-CBC
Users (PCO, PCU and AU)	AES-256-CBC
Keys and key objects (certs)	AES-256-KWP

Table 26 Partition backup data

All the above encryptions use the same ephemeral AES-256 key (Backup session key) which is generated for this purpose. Then, this key is encrypted using AES-256-KWP with a key which is derived from AOKBK and VKBK in case of two-key backup, or additionally PCO_KBK is used in case three-key backup is used.

Every restore operation is carried out with 2FA enabled. If the user is not logged in using 2FA, the restore still proceeds, but the partition is treated as a non-CC partition after the restore. This means that CC mode remains disabled for that partition, even if it was originally created with --enforce-cc and the FIPS state is 2 or 3.

Each restore operation requires a fresh login. For instance, if multiple partitions are being restored within the same session, the user must log out after each restore operation and log back in (with 2FA) before initiating the next one.

Two-factor authentication (2FA) consists of two steps: password verification and signature verification. The signature is generated during login using a combination of the password and a nonce. During user creation, the MCO stores a public key—generated externally by another user outside the TOE—which is later used to verify the signature.

The TOE supports below specified algorithms and key lengths for verifying the signature.

Key type	Key size (bits) / Curve type	Hash Sizes	Algorithm supported
RSA	2048 and 4096	SHA256	pkcs#1 v1.5
ECC	P256, P384 and P521	SHA512	ECDSA

6.4 AUDIT

The TSF includes an audit trail to keep track record of management operations and cryptographic operations performed in the TOE (FAU_GEN.1, FAU_GEN.2). These track uses Octeon RTC to establish and maintain a synchronized real-time clock, employed for time-stamping requirements timestamps to preserve the consistency (FPT_STM.1).

Each log entry has common fields such as Timestamp, Sequence Number, and Opcode. Optional and additional opcode-specific fields such as key info and user info are also described.

Each field is described below:

Field	Description	Size
Timestamp	This is the epoch/unix time, which counts the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970.	64 bits
Per Partition Log Sequence Number	This is made up of: • A 32-bit reboot counter: This is a persistent counter that gets incremented on every reboot of the HSM. This number never gets reset to 0. • A 64-bit log entry sequence number: This is a volatile counter that gets incremented when logging every loggable command. On every reboot of the HSM or when the partition gets destroyed, the sequence number goes back to zero.	96 bits
Opcode	This is the opcode that is used to describe an operation.	32 bits
User ID	User Identification.	16 bits
Response Code	This field shows the result of the API call. For example: • R=SUCCESS • R=ERR_USER_NOT_LOGGED_IN	32 bits
Extra Options (Opcode specific)	This includes multiple key value pairs that are options to the API to behave in one way or the other. This is especially useful for management commands that have a variety of options. For example: NEW_USERNAME=crypto_user3 2 FA_ENABLED=yes (for a CN_CREATE_USER opcode) For crypto operations on a key handle, the KCV for the key will be	A union can be defined with key-value pairs for each opcode. This reflects the size of this field.

Field	Description	Size
	provided as part of the audit logs.	
User Defined Data (UDD)	This is an optional 32-byte input argument specified by the API caller. It gets returned along with the log entry for that API call. When this field is specified by the customer, it takes that value; otherwise, it is zero.	256 bits

Table 27 Logs fields

Audit log is protected (FAU_STG.3). Appliance User is the only role with authorization to retrieve logs. audit removal allows the Appliance User to delete logs from the TOE buffers, while no user is allowed to modify the audit. The default mode is the *block commands mode*, where the commands will be blocked upon log buffers are full. Firmware has 30720 log buffers for each partition. Of these 29718 entries are for regular commands 1000 entries are reserved for TIME_INFO_OPCODE and 2 entries are reserved for performing zeroizeHSM from partition and FinalizeLogs. This is useful when the regular log buffer is full, and the AU was unable to retrieve the logs.

Exporting audit reports is a privilege reserved solely for users with the AU role (FMT_SMR.1). Exporting logs is a functionality that stems from accessing them. Whenever an administrator gains access to the logs, they are automatically exported to the server where the TOE is installed in order to free up space. When retrieving audit logs, the firmware does not immediately clear them to make space for new logs. Instead, they must be acknowledged (ack'ed) in order to be cleared. This means that the same set of logs may be retrieved multiple times before being ack'ed, which can be beneficial for users if the requested logs were not received properly. After a set of logs are retrieved and ACKed, the user can request signature on all these logs. TOE signs the hash of the messages and returns the signature to the application/user. The acknowledgement (ACK) can be sent as an independent notification or combined with other log retrieval commands (FMT_SMF.1, FMT_MTD.1/AuditLog).

6.5 SELF-PROTECTION

The correct function of the TSF is monitored by a set of self-tests (FPT_TST.1) which ensures all cryptographic operations are working properly.

In addition to power-on self-tests, conditional and continuous health tests are run by the TOE. Failure of any of the self-tests causes the module to go into an error state (FPT_FLS.1). If the failure happens during periodic execution of Cryptographic Algorithm Self-Tests (CASTs), module rejects all commands received. It is required to restart the TOE to recover from the situation.

Data output except for status log messages is inhibited during self-tests, zeroization, and error states. Status information does not contain CSPs or sensitive data. The conditional cryptographic algorithm self-tests (CASTs) run periodically. The periodicity is configurable by the MCO and by default runs for every 24 hours. The execution of CAST causes a momentary (less than a second) service interruption.

The voltage monitoring happens continuously by the module, which samples the voltage rails for every 400 micro-seconds. The temperature monitoring happens continuously by the module for every

30 seconds. In case the temperature or voltage reaches a value outside the expected range, the TOE shuts down automatically.

The cryptographic module shall perform the following power up, continuous, and conditional self-tests:

- Pre-operational software/firmware integrity test
  - CRC-32 integrity test
  - Firmware integrity tests. RSA 2048-bit SHA-256 signature verification.
- Pre-operational critical functions tests. The module runs the following critical functions tests, which are required to ensure the correct functioning of the device.
  - Temperature monitor test
  - Voltage monitor test
- Conditional self-tests.
  - Conditional cryptographic algorithm test. The operator is capable of commanding the module to perform the power-up self-test by cycling power or resetting the module. Power-up self-tests do not require any operator action.
    - CPT Library/HW Crypto Algorithms:
      - AES CBC Encrypt & Decrypt KAT (128-bit key)
      - HASH_DRBG KAT (SHA512)
      - RSASP (2048-bit) KAT
      - RSA Encrypt and Decrypt KAT (2048-bit)
      - ECDSA Signature Gen and Verify KAT (P-256 using SHA-256, SHA-384, SHA-512)
      - CMAC KDF in Counter KAT (AES 128-bit key)
      - HMAC KDF in Counter KAT (HMAC-SHA-256)
      - HASH_DRBG KAT (SHA-512)
      - TLS KDF KAT (HMAC-SHA-256)
      - AES CCM KAT (AES 128-bit key)
      - KAS-ECC-SSC KAT (P-256)
    - SWCrypto Algorithms:
      - AES CBC Encrypt & Decrypt KAT (128-bit key)
      - ECDSA PKV KAT (P-256)
      - CTR_DRBG KAT (AES 256-bit)
      - ECDSA Sig Gen and Sig Ver KAT (P-256 with SHA-1, SHA-256, SHA-384, SHA-512)
      - RSA Encrypt and Decrypt KAT (2048-bit)
      - RSA Sig Gen, Sig Ver KAT (2048-bit)
      - HMAC KDF KAT (HMAC-SHA-256)
      - X9.63 KDF KAT (SHA-256)
      - HMAC-SHA2-256 KAT (HMAC-SHA-256)
      - TLS KDF KAT (HMAC-SHA-256)
      - rev3 KAS KAT (P-521 and HMAC-SHA-512)
      - rev3 KAS-ECC-SSC KAT (P-256)
      - AES Key Wrap and Unwrap KAT (AES 128-bit)
      - rev2 KAS KAT (RSA 2048-bit)

- rev2 One-Step KDF and Two-Step KDF KAT (SHA-224)
- AES CMAC KAT (AES 128-bit)
- CMAC KDF KAT (AES 128-bit)

- LS2_UBOOT Crypto Algorithms Library:
  - RSA Sig Ver (2048-bit) KAT with SHA-256
- OCTEON HW RNG:
  - SP 800-90B startup health tests (RCT and APT)
  - SP 800-90B (HW RNG) continuous health tests (RCT and APT)
- DRBG
  - SP 800-90A Health Tests (AES CTR 256-bit), (Hash SHA512)

- Conditional pair-wise consistency test
  - ECDSA/ECDH Pairwise Consistency Test (ECDSA FIPS 186-5)
  - RSA Pairwise Consistency Test
- Conditional software/firmware load test (7.10.3.4)
  - Firmware load test (RSA Signature Verification)
  - RSA 2048-SHA-512
- Conditional critical functions test (7.10.3.7)
  - Temperature monitor test
  - Voltage monitor test
- Periodic self-tests
  - Module performs periodic self-tests for a configured duration

In the case they don't reach an adequate operational state, the TSF can manage secure failure state. If a failure is detected, the TOE reacts differently depending on the type of event that triggered the event. In accordance with FPT_FLS.1, if the type of event is (2), the TOE automatically turns off and is not restarted until the direct intervention of a user. If the type of event is one of (1), (3) or (4), the state of the TOE changes to a non-operational state in which only the following list of read-only operations are allowed:

- Open session
- Close session
- Initialize application
- Finalize application
- Get token information
- Get version
- Get partition information
- Get all partition information
- Application cleanup

The security boundary protects the secrets of the HSM from alteration. The security boundary of the board is an epoxy shell placed on both sides of the printed circuit board that serves as a layer of protection. The Epoxy shell covers the PCB and circuits inside with epoxy that will severely damage the HSM if the shell is forcibly removed and allow detection of tampering attempts (FPT_PHP.1).

The modification of the voltage and temperature outside the normal operating range or the activation of tamper pin through the proper signal will activate the zeroization of the device (FPT_PHP.3).

The TOE enforces integrity protection on all keys. The integrity mechanism protects both the key value and its security attributes. It consists of a SHA256-based value calculated over the whole key object (key value + security attributes), which is verified prior allowing any operation on (or with) the key. In the event integrity verification fail, the module prohibits the targeted operation and returns an error message (FDP_SDI.2). The event is also notified through a record in the audit log (FAU_GEN.1).

6.6 SECURE CHANNELS

The end-to-end encryption feature of the TOE allows an application to initiate a TLS connection with the TOE to ensure the confidentiality and integrity of the data communicated to the HSM and the authentication of the endpoint it is connected to.

The connection is based on TLS v1.2 following [TLS1.2] specification with the cipher-suite TLS-ECDHE-RSA-AES256-GCM-SHA384 with curve secp384r1 (FTP_TRP.1/Local, FTP_TRP.1/External).

The TOE acts as the server and the host application acts as client. The server private key is the partition private key (PAK), which is generated for each partition when it is created. The server certificate used for the TLS connection is the partition certificate (PAC) and optionally with the mTLS (Mutual TLS), client certificate is used to authenticate the client that is connecting to the TOE. The end-to-end encryption feature is enabled using the initialization configuration parameters. Once this feature is enabled, specific commands are encrypted except those allowed to be executed over a non-secure channel.

The following operations cannot use secure channel:

- Change password (1)
- Logout (1)
- Get object information (1)
- Set node ID (1)
- Extract masked object (1)
- Insert masked object (1)
- Extract masked object user (1)
- Insert masked object user (1)
- Find all objects (1)
- Find all objects using counter (1)
- Create user (2)
- Generate password encryption key (2)
- Get request for certificate authentication
- Get certificate for certificate authentication
- Store certificate for certificate authentication
- Get partition audit logs
- Get partition audit details
- Get partition audit signature
- Acknowledge audit logs

- Delete tombstone object
- Get partition single key handle hash
- Get partition administrative key handles hash
- Get token information
- Get partition information
- Delete user
- Get user list
- Get session information
- Login
- Get version
- Get login failure counter
- Close all sessions
- Get M value
- Get token
- Find all objects using counter
- Get source random numbers
- Authenticate source in key exchange
- Authenticate target in key exchange
- Get KBK slot information
- Set KBK as primary
- Get policy set

(1) Only when AU is performing this operation, cannot use secure-channel.

(2) During HSM initialization, there is no E2E established so this opcode can be used during that time. Once HSM initialization is complete, this command is forced to use secure-channel.

All communications focused from the TOE side are local in nature, with the server being the end point of the communication. The physical interface through which all communications reach the TOE is the same, regardless of the origin, and is based on the PCIe interface interaction between the host and the TOE.

Local communications are used to manage the Master Partition through the PF of the PCIe (FTP_TRP.1/Local). Note that this local communication within the same local server or appliance does not require the secure-channel, since the local environment provides sufficient protection.

Remote communications are reserved to user partitions and are based on PCIe VFs on the Host where ToE is connected (FTP_TRP.1/External) and requires the establishment of a secure-channel.

6.7 AUTHENTICATION & AUTHORIZATION

The TOE includes several roles for distinguishing management functions (FMT_SMR.1/Roles, FMT_SMF.1). These roles are identified and authenticated before the TOE can be managed given an identity-based operator authentication composed by username and password (FIA_UID.1, FIA_UAU.1).

Role	Description
MCO	**Master Crypto Officer** The Master Crypto Officer role (MCO) is allowed only on the master partition, which is mandatory to use the HSM. This role has access to administrative services offered by the module or HSM. This role is used to configure non-master partitions (create, provision, resize, and delete) but cannot access their resources (e.g., cannot manage or use non-master partition keys).
Pre-CO	**Pre-Crypto Officer** This role is an optional role with limited functionality; eventually transitions into PCO. During partition initialization, default credentials are used to create a Pre-CO or a PCO. The Pre-CO is a restricted role primarily for configuring certificates and setting up a PCO. After a PCO is set up for a partition, the Pre-CO role is no longer accessible.
PCO	**Partition Crypto Officer** This role has access to administrative services of the partition and can configure PCU and AU identities. The HSM supports more than one Crypto Officer role with a requirement there shall be at least one.
PCU	**Partition Crypto User** This role has access to all cryptographic services offered by the partition; its purpose is operational use of the module
AU	**Appliance User** This role has access to partition audit logs. It is used to set up and synchronize clusters.

Table 28 Roles privileges

The set of read-only commands that can be executed prior to user authentication, based on the type of partition, are the following:

Partition	Command	Description
Master partition	Get HSM label	Returns LS2 HSM board label string.
	Get HSM diagnosis info	Returns LS2 HSM diagnostics (Resources [Memory, Accelerator engines, Firmware processes] status / health, statistics, configuration, HSM/RTC Time).
	Get firmware version	Returns Firmware version
User partition	Get token info	Returns token information: (Resources [Memory, Accelerator engines, Firmware processes] status / health, statistics, HSM/RTC Time, HSM

Partition	Command	Description
		Partitions, HSM serial no, label, FW version, Bootloader version, infra keys fingerprint).
	Get partition info	Return information about one or all partition: configuration, audit logs status, Partition name, resource statistics.
	Get user list	Return list of users on the partition.

Table 29 Read-only commands for unauthorized users

- Users access can be blocked based on a configurable authentication attempt (FIA_AFL.1). Once users are blocked, they can be unblocked by a higher-level administrator as defined in Table 28 (FMT_MTD.1/Unblock).

6.8 PARTITION RESOURCES ACCESSIBILITY

The TOE is divided in several partitions as security domains, where they are constrained the access to the resources (FDP_IFC.1/Partitions, FDP_IFF.1/Partitions).

Partitions access their allocated resources such as the CPT cryptographic processor and the allocated RAM area in isolation. Partition User Keys, CSPs and their attributes are implemented via a database hosted. And User Identities, credentials and roles are also implemented via a database hosted. The database (eMMC) is exclusive to each partition. Partition Isolation, authentication and enforce authorization for the stake holder entities defined in the TOE as per their roles and responsibilities (User Partitions(s), Master Partition). Trusted Micro Services (TMSes) provide isolation of assets (CSPs, keys, certificates) and minimize their footprint(copies) in various daemons, their address spaces.

Access to CPT cryptographic processor

The CPU has a multi core/engine crypto co-processor, called CPT, embedded within the chip. CPT has 32 engines for asymmetric cryptography and an additional 96 engines for symmetric cryptography. These engines are stateless and run Marvell inhouse developed microcode contained in the firmware of the TOE. Microcode does not maintain any state across multiple operations. All requests/operations, including multi-step, will carry the context required to process the request through RAM, read data, write results etc.

CPT engines are accessed using 64 hardware queues exposed via virtual functions. A virtual function can be assigned with one or more hardware queues to offload crypto operations to CPT. Crypto offload operations from a hardware queue can be processed by one or more free engines. Each partition will have a dedicated command queue to offload requests to CPT but there is no dedicated CPT core assigned to a partition.

CPT is flexibly divided into 168 logical cores and the logical cores are assigned to a virtual function. LS2 HSM assigns one virtual function with one dedicated hardware queue per partition. A strict throttling mechanism is implemented to maintain the fair-share between Partitions. CPT cores are shared among partitions.

Access to RAM

LS2 HSM has 16GB RAM which is divided and assigned to each partition. Linux Cgroup policies confine one Partition to its assigned RAM limits. ARM's SMMU is configured to ensure the assigned RAM is isolated from the neighboring partitions.

On bootup, HSM firmware would load the Partition Content (Policies, Users and Keys), decrypt and maintain in memory for performance optimizations.

7 ACRONYMS

The following table shows the acronyms used in the evaluation.

Acronym	Meaning
AES	Advanced encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
APT	Adaptative Proportion Test
AU	Appliance user
CC	Common Criteria
CCA	CPU Core Access Control
CPU	Central Process Unit
CSP	Critical Security Parameter Crypto Service Provider Certification Service Provider
DRBG	Deterministic random number generator
DTBS/R	Data To Be Signed or its unique representation
EAL	Evaluation Assurance Level
ECC	Error Correction Code / Elliptic Curve Cryptography
ECDH	Elliptic-Curve Diffie-Hellman
ECDSA	Elliptic-Curve Digital Signature Algorithm
eMMC	embedded Multimedia Card
FIPS	Federal Information Processing Standard
F-RAM	Ferro-electric non-volatile RAM
H.A.	Hierarchically above
HSM	Hardware Security Module
HW	Hardware
ID	Identifier
IPC	Inter-process Call
IT	Information Technology
KDF	Key Derivation Function
MCO	Master Crypto Officer
MCU	Memory Control Unit
NIST	National Institute of Standards and Technology
OSP	Organizational Security Policy
PAC	Partition Certificate
PAK	Partition Private Key

Acronym	Meaning
PCI	Payment Card Industry
PCIe	Peripheral Component Interconnect Express
PCO	Partition Crypto Officer
PCU	Partition Crypto User
PF	Physical Function
PID	Process identifier
POTAC	Partition Owner Trust Anchor Certificate
PP	Protection Profile
RAD	Reference Authentication Data
RCT	Repetition Count Test
RGB	Red Green Blue
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman
SCD	Signature Creation Data
SFP	Security Function Policy
SMBus	System Management Bus
SP	Special Publication
SR-IOV	Single-root input/output virtualization
SSL	Secure Socket Layer
ST	Security Target
SVD	Signature Verification Data
TBD	To Be Determined
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFi	TSF Interface
TSP	Trusted Service Provider
UART	Universal Asynchronous Receiver-Transmitter
UCD	Universal Controller Device
VAD	Verification Authentication Data
VF	Virtual Function

Table 30 Abbreviations

8 GLOSSARY OF TERMS

For the purposes of this document, the acronyms, terms and definitions given in the documents referenced in 9 apply.

Additional terms defined for the purposes of this evaluation are listed below.

Term	Meaning
Assigned Key	A key (usually a secret key) with the 'Assigned Flag' attribute set to 'assigned', meaning that: • The 'Reauthorisation conditions' and 'Key Usage' attributes cannot be changed. • The Authorisation Data attribute can only be changed by presentation of the current Authorisation Data – it cannot be changed or reset by an Administrator. • the key cannot be imported or exported. These properties of an Assigned Key support the sole control of a key that is required for secret keys used to create digital signatures.
Augmentation	Addition of one or more requirement(s) to a package
Authorization Data	Data, including data particular to the user, which is used to control access to (and thus use of) a key. Data particular to the user may include data derived from a secret known only by the user, data derived from a device held by the user and/or data derived from biometric features of the user. Other parts of the Authorization data may include data held within the cryptographic module, data held by administrator(s) or data provided by the application.
Cryptographic module	Set of hardware, software and firmware used to generate the Subscriber-SCD/Subscriber-SVD pair and which represents the TOE.
Data to be signed	The complete electronic data to be signed, such as QC content data or certificate status information.
Data to be signed representation	The data sent to the TOE for signing which is (a) a hash-value of the DTBS or (b) an intermediate hash-value of a first part of the DTBS and a remaining part of the DTBS or (c) the DTBS itself. The client indicates to the TOE the case of DTBS-representation, unless implicitly indicated. The hash-value in case (a) or the intermediate hash-value in case (b) is calculated by the client. The final hash-value in case (b) or the hash-value in case (c) is calculated by the TOE.
Digital signature	Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of that unit and protect against forgery e.g. by the recipient.

Term	Meaning
Electronic Seal	Data in electronic form which is attached to or logically associated with other data in electronic form to ensure the latter's origin and integrity.
Electronic Timestamp	Data in electronic form which binds other data in electronic form to a particular time establishing evidence that the latter data existed at that time.
Evaluation Assurance Level	Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package
Operational Environment	Environment in which the TOE is operated
Protection Profile	Implementation-independent statement of security needs for a TOE type
Reference Authentication Data	Data persistently stored by the TOE for verification of the authentication attempt as authorised user.
Secret Key	Is the key to be protected by the user to ensure the confidentiality of the encrypted data, which is both the key used in symmetric cryptographic functions and the private key used in asymmetric cryptographic functions.
Security Target	Implementation-dependent statement of security needs for a specific identified TOE
Signature-creation data	Unique data, such as codes or private cryptographic keys, which are used by the signatory to create an electronic signature.
Signature-verification data	Data, such as codes or public cryptographic keys, which are used for the purpose of verifying an electronic signature.
Target Of Evaluation	Set of software, firmware and/or hardware possibly accompanied by guidance
Trust Service	Electronic service which enhances trust and confidence in electronic transactions. NOTE: Such trust services are typically but not necessarily using cryptographic techniques or involving confidential material.
Universal Controller Device	Monitors power rails and thermal events; sets GPIOs to reset OCTEON and turn on LEDs.
User data	Data created by and for the user that does not affect the operation of the TOE Security Functionality (TSF).
Verification authentication data	Authentication data provided as input by knowledge or authentication data derived from user's biometric characteristics.

Table 31 Glossary of terms

9 DOCUMENT REFERENCES

The following table shows the documents referenced in this evaluation.

Reference	Document
[ANSIX9.63]	Key Agreement and Key Transport Using Elliptic Curve Cryptography. Rev. 2017
[CC2022P1R1]	Common Criteria for Information Technology Security Evaluation, Version 2022, Revision 1, Part 1: Introduction and general model
[CC2022P2R1]	Common Criteria for Information Technology Security Evaluation, Version 2022, Revision 1, Part 2: Security functional components
[CC2022P3R1]	Common Criteria for Information Technology Security Evaluation, Version 2022, Revision 1, Part 3: Security assurance components
[CC2022P4R1]	Common Criteria for Information Technology Security Evaluation, Version 2022, Revision 1, Part 4: Security assurance components
[CC2022P5R1]	Common Criteria for Information Technology Security Evaluation, Version 2022, Revision 1, Part 5: Security assurance components
[CEM2022R1]	Common Criteria Evaluation methodology, Version 2022, Revision 1
[CEN EN 419221-5]	Protection profiles for TSP Cryptographic modules - Part 5 Cryptographic Module for Trust Services. 2018 E Version 1.0
[FIPS 186-5]	Digital Signature Standard (DSS). NIST. February 2023
[FIPS 197-upd1]	Federal Information Processing Standards Publication 197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). Published November 26, 2001; Updated May 9, 2023
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC). NIST. July 2008
[FIPS 202]	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. NIST. August 2015
[HMAC]	RFC 2014. IETF. October 1996
[HW-INS-GUID]	LS2-HSM-Adapter-Hardware-Installation-Guide_Rev.3
[PKCS#1]	RFC3447. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
[Regulation]	REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing. Directive 1999/93/EC and Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework
[SOG-IS-Crypto]	SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, v1.3, February 2023

Reference	Document
[SP 800-108r1-upd1]	Recommendation for Key Derivation Using Pseudorandom Functions August 2022.
[SP800-135r1]	Recommendation for Existing Application-Specific Key Derivation Functions. NIST. Rev. 1. December 2011
[SP800-38A]	Recommendation for Block Cipher Modes of Operation: Methods and Techniques. NIST. December 2001.
[SP800-38B]	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. October 6, 2016
[SP800-38C]	Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. NIST. May 2004
[SP800-38D]	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST. November 2007
[SP800-38F]	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping. NIST. December 2012
[SP800-56Ar3]	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. Rev. 3
[TS 119 312]	ETSI TS 119 312 Electronic Signatures and Infrastructures (ESI); Cryptographic Suites v1.2.1, 2017-05
[TLS1.2]	RFC, The Transport Layer Security (TLS) Protocol, Version 1.2. August 2008 (https://www.ietf.org/rfc/rfc5246.txt)
[LS2 Boot Loader]	MARVELL-LS2-UBOOT-10.24-0702-R01-SB MARVELL-LS2-UBOOT-10.24-0702-R02-SB
[Crypto Functional Requirements Spec]	CCDB-018-v1.0-2025-Jan-31-Final-Specification_of_Functional_Requirements_for_Cryptography.pdf
[Errata]	Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, dated 22 July 2024
[Transition Policy]	Transition Policy to CC:2022 and CEM:2022, CCMC-2023-04-001, dated 20 April 2023

Table 32 List of document references

Note: SOG-IS refers to the standards that have been withdrawn, hence it is updated to the latest version.