Rambus

Security IP
SESIP Security Target for PSA Certified RoT Component Level 2

CMRT RT-634

Based on SESIP methodology, version “Public release 1.2”

Document Revision: F
Document Date: 2024-02-20
Document Number: 001-634220-503/941

Document Status: Accepted

Rambus Inc. Corporate Headquarters
4453 North First Street, Suite 100
San Jose, CA 95134
Phone: +1 408-462-8000
Website : https://www.rambus.com/
Contact : sipsupport@rambus.com

Rambus ROTW Holding B.V.
Boxtelseweg 26A
5261 NE Vught
The Netherlands
Phone: +31-73-6581900

Table of Contents

List of Tables

List of Figures

Document Revision History

Doc Rev	Date (Y-M-D)	Author	Purpose of Revision
A	2023-10-13	MWANG	Creating Draft
B	2023-12-19	MWANG	Reviewed by JPW and processed the comments; Updated references versions; Add Secure communication SFRs.
C	2024-01-30	MWANG	Add Secure Debugging SFR Add iteration for Secure Communication Revised according to lab's comments
D	2024-02-05	MWANG	Add access control SFRs Update SOE
E	2024-02-08	MWANG	Change section 1.8.5
F	2024-02-20	MWANG	Update after EM1, based on Certifier's comments

1 Introduction

The Security Target describes the Platform (in this chapter) and the exact security properties of the Platform that are evaluated against [SESIP] (in chapter “Security requirements and implementation”) that a potential consumer can rely upon the product upholding if they fulfil the objectives for the environment (in chapter “Security Objectives for the operational environment”).

1.1 SESIP Profile reference

Reference	Value
PP Name	SESIP Profile for PSA Certified RoT Component Level 2
PP Version	1.0 REL 02
Assurance Claim	SESIP Assurance Level 2 (SESIP 2)
Optional and additional SFRs	Secure communication Support Secure communication Enforcement Secure External Storage Field Return of Platform Residual Information Purging Authenticated Access Control

1.2 ST reference

See title page.

1.3 Platform reference

Reference	Value	
Platform Name	*CMRT RT-634 Root of Trust Core*	
Platform Version	*RT-634 v2.2*	
Platform Identification	*Hardware*	*60020723*
	Software	*2024-01-18-ga7c02de*
Platform type	*Security Soft IP*	

Table 1 Platform Reference

1.4 Included guidance documents

The following documents are included with the platform.

Reference	Name	Version
[SYS_ARC]	*CMRT System Architecture and Users Guide*	*Rev A*
[HW_Manual]	*CMRT_External_Ref_Spec_RevD_Superset_Build_10-20-2023*	*Rev D*
[API]	*2023-12-11-RT-6xx-FIPS-140-3-eSW-v2.02*	*V2.02*
[HW_INT]	*CMRT Integration Guide*	*Rev D*
[CONFIG]	*CMRT Configuration Details*	*Date: October 20. 2023*
[SEC_GUID]	*CMRT_User_Security_Guidance_Manual*	*Rev A*

Table 2 Included guidance documents

1.5 Acronyms

BNAK	Builder Netlist Keysplit
CPU	Central Processing Unit
ECC	Error-Correcting Codes
Fboot	First-stage Bootloader
FW	Firmware
HUK	Hardware Unique Key
HW	Hardware
IP	Intellectual property
KAT	Known Answer Test
KTC	Key Transport Core
KWP	Key Wrap With Padding
MPU	Memory Protection Unit
NVM	Non-Volatile Memory
OTP	One Time Programmable
PNAK	Perso Netlist Keysplit
PSA	Platform Security Architecture
PSIRT	Product Security Incident Response Team
RAM	Random-access Memory
ROM	Read-only Memory
Sboot	Second-stage Bootloader
SNAK	SOC Netlist Keysplit
SoC	System on Chip
TOE	Target of Evaluation
TRNG	True Random Number Generator

1.6 Document references

Reference	Name	Version
[SESIP]	*SESIP methodology*	*Version 1.2*
[SEC_DLV]	*Security IP Secure Delivery Process*	*Version 1.0*
[FLR]	*Rambus Vulnerability Management Procedure*	*Revision: B*
ATE	*Functional Testing document and evidence*	-

1.7 (Optional) Other Certification

Not applicable

1.8 Platform functional overview and description

1.8.1 Platform type

The Rambus CMRT RT-634 Root of Trust IP are fully programmable FIPS 140-3 compliant hardware security cores with optional Quantum Safe security by design for data center, AI/ML, as well as general purpose semiconductor applications. They protect against a wide range of hardware and software attacks through state-of-the-art anti-tamper and security techniques.

1.8.2 Physical Scope

The TOE has hierarchical secure execution environment which separated into 3 levels:

1. Security Monitor: highest privilege
2. Zephyr: supervisor
3. Container: user process

The TOE scope is depicted in Figure 1. The boundary is represented within the red line box. The white boxes represent the CMRT components that comprise the IP cores (the CMRT firmware is stored in Program ROM and Program RAM). The yellow boxes represent the components that are provided in the IP core but must be replaced or adjusted during the synthesis process as they are technology dependent (OTP, SRAM, TRNG FROs).

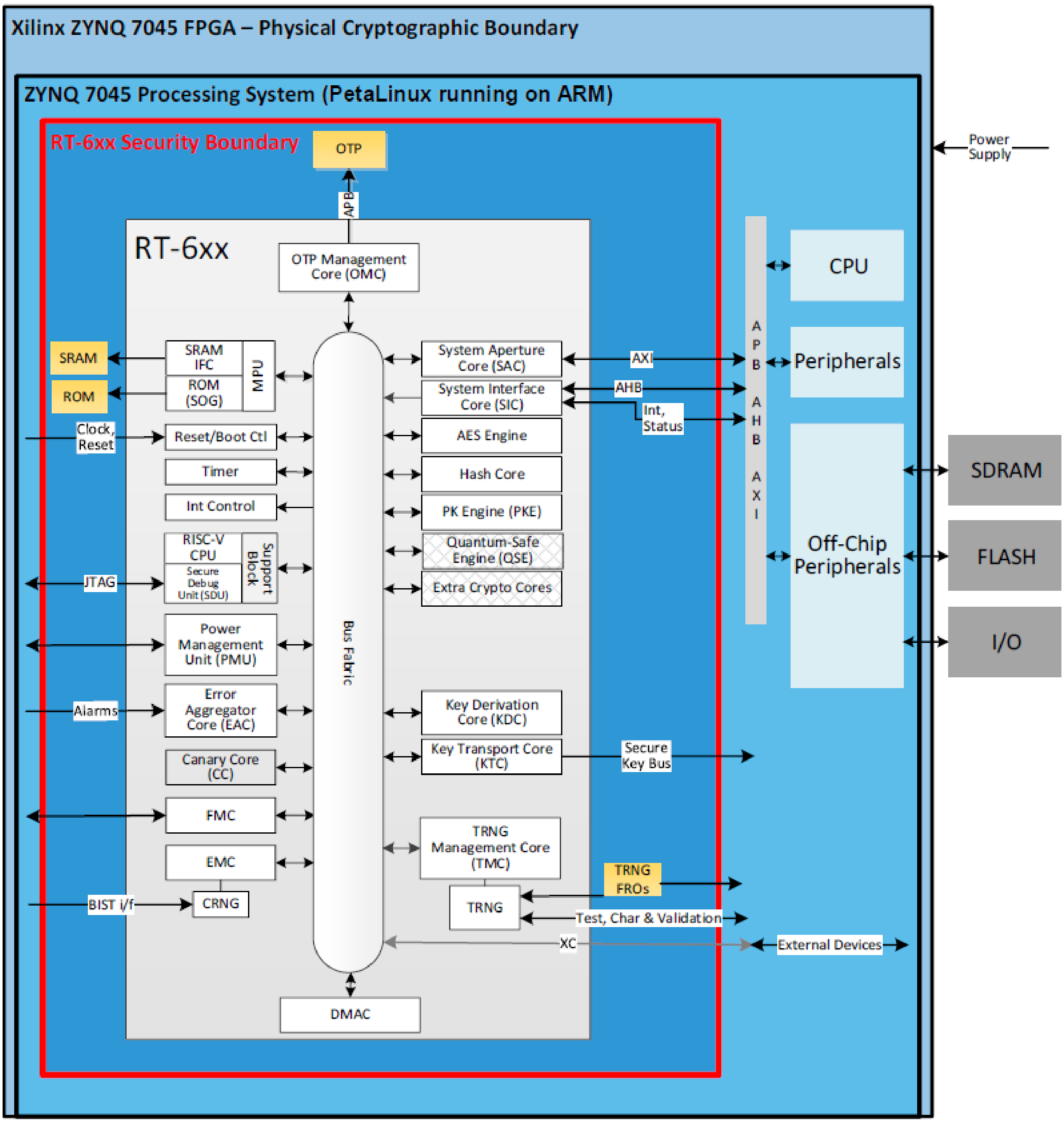


Figure 1 TOE scope

1.8.3 Logical Scope

The logical scope includes:

- First-Stage Boot loader (FBoot)
- Second-Stage-Boot loader (SBoot)
- Supervisor Application (supported by the CMRT Supervisor Software Development Kit)
  - The supervisor application supports the execution of the full suite of KATs, setting the FIPS mode indicator, user controls and FIPS mode services.
  - The FIPS mode is set when all KAT have passed. This means the TOE is in FIPS 140-3 Approved mode.

The TOE allows 2 roles: Crypto officer and normal users (maximum 6 users). Both should be authenticated before sending any commands. Anonymous logins are not supported.

1.8.4 Usage and Major Security Features

The main security features of the TOE are as follows:

- Secure asset management
- Secure booting
- Self-isolated from other parts
- Cryptographic functions
- Secure Firmware update
  - Anti roll-back
- Random Number Generator
- Secure storage of security parameters including keys
- Support of isolation of the platform
- Secure Debug

1.8.5 Required Hardware/Software/Firmware

- External NVM for persistent storage.
- Implementation of OTP and SRAM (technology dependent).

2 Security Objectives for the operational environment

For the platform to fulfill its security requirements, the operational environment (technical or procedural) shall fulfil the following objectives.

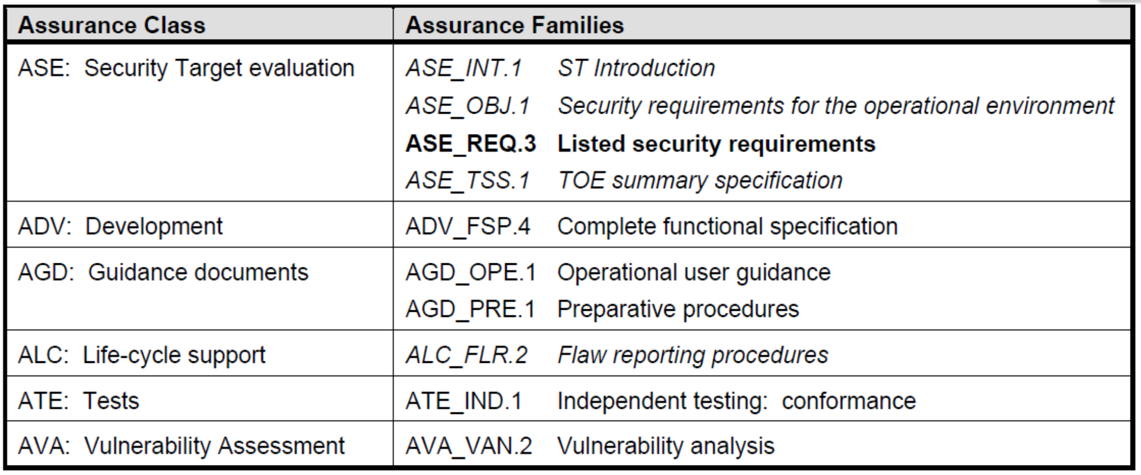
ID	Description	Reference
INTEGRATION	In order to build the product securely, the hardware integration manual [HW_INT] shall be followed.	[HW_INT] all sections
SELF_TEST	In order to ensure Power-On Self-Tests are executed, follow the security guidance.	[SEC_GUID] section 2.1
SECURE _PROVISION	Relevant keys shall be provisioned securely into OTP during manufacturing.	[SEC_GUID] section 3.2
VERSION_UPDATE	When update is available, the chip vendor shall update the TOE version information.	[SEC_GUID] section 4.2
APPROVED_ALGORITHMS	Users should use NIST approved cryptographic algorithms.	[SEC_GUID] section 4.6, 4.7
RNG_CONFIG	TRNG and DRBG configuration shall follow the security guidance.	[SEC_GUID] section 3.3, 3.4
RNG_USE	TRNG and DRBG use shall follow the security guidance.	[SEC_GUID] section 4.3, 4.4
KEY_MANAGEMENT	Cryptographic keys and certificates outside of the platform are subject to secure key management procedures.	This document
TRUSTED_USERS	Actors in charge of platform management, for instance for signature of firmware update, are trusted.	This document
UNIQUE_ID	The integrity and uniqueness of the unique identification of the platform must be provided by the platform user during the personalization stage	[API] [SEC_GUID] section 4.1

3 Security requirements and implementation

3.1 Security Assurance Requirements

The claimed assurance requirements package is: SESIP2 as defined in [SESIP].

The assurance requirements are shown below:



Assurance Class	Assurance Families	
ASE: Security Target evaluation	*ASE_INT.1*	*ST Introduction*
	ASE_OBJ.1	*Security requirements for the operational environment*
	ASE_REQ.3	**Listed security requirements**
	ASE_TSS.1	*TOE summary specification*
ADV: Development	ADV_FSP.4	Complete functional specification
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: Life-cycle support	*ALC_FLR.2*	*Flaw reporting procedures*
ATE: Tests	ATE_IND.1	Independent testing: conformance
AVA: Vulnerability Assessment	AVA_VAN.2	Vulnerability analysis

Table 3 SESIP2 Assurance Requirements

3.1.1 Flaw Reporting Procedure (ALC_FLR.2)

In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2), including a process to generate any needed update and distribute it, the developer has defined the following procedure:

Rambus has built a Product Security Incident Response Team (PSIRT), which is responsible for responding to security incidents. PSIRT manages receipt, investigation and releasing of information about security issues regarding Rambus products.

For external parties that that wish to report a vulnerability, they may contact Rambus via the link below:

https://www.rambus.com/security/response-center/report-vulnerability/

See [FLR] for details.

The firmware version is formed by “yyyy-mm-dd-xxxxxxxx”. The last 8 digits is the tag from the hash of the commit for creating the release. Products that have a smaller date number than the firmware or smaller hardware version number indicated in section 1.3, are considered as older versions. E.g. 2023-12-19-xxxxxxxx, 60020720.

Products that have a larger number than the hardware and firmware version number indicated in section 1.3, is considered as newer versions. E.g. 2024-02-19-xxxxxxxx, 60020724.

3.2 Base PP Security Functional Requirements

As a base, the platform fulfils the following security functional requirements:

3.2.1 Verification of Platform Identity

The platform provides a unique identification of the platform, including all its parts and their versions.

Conformance rationale:

The platform ID can be read by using “show status” service. Both the hardware and software version will be output. For more details please refer to [API] section 7.6.37.

3.2.2 Secure Update of Platform

The platform can be updated to a newer version in the field such that the integrity, authenticity and confidentiality of the platform is maintained.

Conformance rationale:

The TOE has a secure update mechanism which is similar to secure booting. For details see 3.3.1 . The fboot and sboot is not updatable. Fboot is in the ROM memory space. Supervisor Application is updatable. The TOE fboot verifies the version of all loaded images as a protection from downgrade attacks. See [SYS_ARC] section "OTP Memory Layout".

3.3 SFRs for PSA-RoT Component

3.3.1 Secure Initialization of Platform

The platform ensures its authenticity and integrity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to error state.

Conformance rationale:

CMRT contains fboot and sboot. Fboot is the first-stage bootloader. It is the first code executed by CPU after booting. Sboot is the second-stage bootloader.

Below is the booting flow:

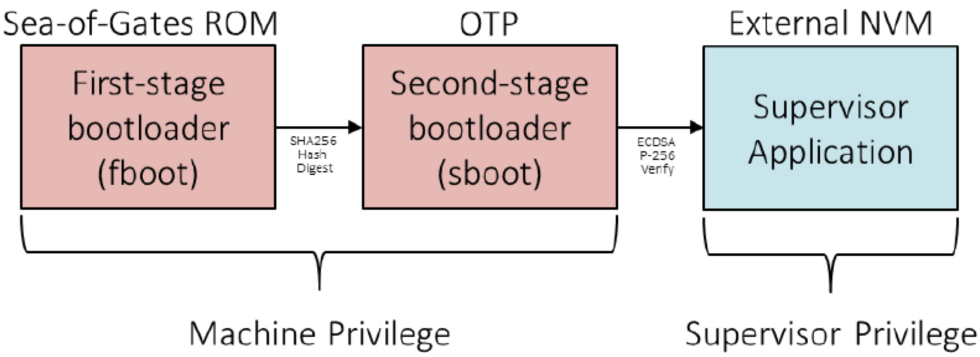


Figure 2 booting flow

- Fboot first copies the sboot image from OTP to internal SRAM.
- (FIPS mode) Fboot executes a SHA256 KAT
  - If failed, fboot writes the error to the register and halts.
- Fboot then computes the SHA256 hash digest of the sboot image in OTP and compares the computed hash digest with the hash digest written with the sboot image in OTP.
  - If the hash digest does not match the value stored in OTP, fboot writes an error to the register and halts.
- Fboot calls the sboot entry point in SRAM
- Sboot copies the next boot image from external NVM to SRAM and verifies the ECDSA signature of the image
- (FIPS mode) sboot executes 2 cryptographic algorithms, SHA256 and ECDSA to verify the supervisor image.
  - Sboot must perform a KAT for each algorithm. If a KAT fails, an error code is written to register and sboot halts.

  - Supervisor application could be encrypted while loading, using AES GCM. See [AYA_ARC] section "Encrypted Image Support" for details.
- The MPU rules (i.e. read-only, read/write etc.) are written to registers for each region of ROM and SRAM. The MPU rules are established either by compile-time variables for the ROM or a footer in the supervisor's image that describes where the .text, .read-only, etc. sections are located.
- Once the MPU is setup, the supervisor execution environment is finalized, and a context switch is performed from machine to supervisor privilege. The CPU begins executing at the supervisor entry point that was setup during the s-mode execution environment setup.

3.3.2 Software Attacker Resistance: Isolation of Platform

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other functional requirements.

Conformance rationale:

The TOE is a Root of Trust for a Chip or SoC. It is isolated from other parts of the Chip or SoC and does not share memory with the Chip or SoC.

3.3.3 Cryptographic Operation

The platform provides the application with operations, and functionality with algorithms as specified in Specifications, key lengths and modes are described in **Table** 4.

Conformance rationale:

Algorithms	Key lengths (bits)	Modes	Specifications	Usage
AES	128, 192, 256	ECB, CBC, CTR, CFB128	NIST FIPS 197 SP800-38A	Encryption, Decryption
	128, 192, 256	CCM	NIST FIPS 197 SP800-38C	Encryption, Decryption
	128, 192, 256	GMAC	NIST FIPS 197 SP800-38B, SP800-38D	MAC Generation and Verification
	128, 192, 256	GCM	NIST FIPS 197 SP800-38D	Encryption, Decryption
	128, 192, 256	CMAC	NIST FIPS 197 SP800-38B	MAC Generation and Verification
	128, 192, 256	KWP	NIST FIPS 197 SP800-38F, RFC3394, RFC5649	Key Wrap/Unwrap
ECDSA	P-224, P-256, P-384, P-521	Key pair generation mode	SP800-56A NIST FIPS 186-4 NIST SP800-186	Key Generation, Key Verification
ECDSA	P-224, P-256, P-384, P-521	Hash Algorithm: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	SP800-56A NIST FIPS 186-4 NIST SP800-186	Signature Generation, Signature Verification

Algorithms	Key lengths (bits)	Modes	Specifications	Usage
				selecting Hash Core 2[1]
ECDSA	P-224, P-256, P-384, P-521	Hash Algorithm: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512	SP800-56A NIST FIPS 186-4 NIST SP800-186	Signature Generation, Signature Verification selecting Hash Core 1[2]
KAS-ECC-SSC	P-224, P-256, P-384, P-521	ephemeralUnified: KAS Role: initiator, responder	SP800-56Arev3	Shared Secret Computation
KAS-ECC	P-224, P-256, P-384, P-521	Function: Full Validation Scheme: ephemeral Unified: KAS Role: Initiator, Responder KDF Methods: with One-Step and Two-Step KDF, MAC Modes: HMAC-SHA-256	SP800-56Arev3 SP800-56Crev3	Key Agreement
HMAC	112-512	SHA-224	NIST FIPS 198-1 NIST 180-4 NIST FIPS 202	MAC Generation, MAC Verification
	126-512	SHA-256		
	192-1024	SHA-384		
	256-1024	SHA-512		
	112-1152	SHA3-224		
	128-1088	SHA3-256		
	192-932	SHA3-384		
	256-576	SHA3-512		
KBKDF	8- 4096 bits Increment 8	Counter mode using HMAC-SHA-256 as PRF	SP800-108 NIST FIPS 198-1 SP800-38B	Key Derivation
KDF	Derived Key Length: 256 Shared Secret Length: 256-512 Increment 128	Counter mode (one step, two steps) using	SP800-56CRev2 NIST FIPS198-1	Key Derivation

[1] Hash Core (HC) 2 implements the following Hash Algorithms SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 and corresponding HMAC.

[2] Hash Core (HC) 1 implements the following Hash Algorithms SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 and corresponding HMAC.

Algorithms	Key lengths (bits)	Modes	Specifications	Usage
		HMAC-SHA-256 as PRF	SP800-38B	
RSA	Modulus: 2048, 3072, 4096	Probable prime with standard key and CRT key format	NIST FIPS 186-4	Key Generation
	Modulus: 2048, 3072, 4096 with SHA-224, SHA-256, SHA-384, SHA-512, SHA2-512/224, SHA2-512/256	RSA PKCSPSS	NIST FIPS 186-4	Signature Verification, Signature Generation, selecting Hash Core 2
	Modulus: 2048, 3072, 4096 with SHA-224, SHA-256, SHA-384, SHA-512	RSA PKCSPSS	NIST FIPS 186-4	Signature Verification, Signature Generation
	modulus: 2048, 3072, 4096 with SHA3-224, SHA3-256, SHA3-384, SHA3-512	RSA PKCSPSS	NIST FIPS 186-4	Signature Verification, Signature Generation
CKG	128, 192, 256	AES key (modes ECB, CBC, CTR, CFB128, GMAC, CMAC, KWP)	SP800-133Rev2	Cryptographic Key Generation
	112 – 512 bit	HMAC key	SP800-133Rev2	Cryptographic Key Generation
	2048, 3072, 4096	RSA key pair	SP800-133Rev2	Cryptographic Key Generation
	P-224, P-256, P-384, P-521	ECDSA/EC Diffie-Hellman key pair	SP800-133Rev2	Cryptographic Key Generation
KTS-OAEP	n=(1024 to 3072)	RSA-OAEP	SP800-56B	Key Transport Scheme
RSA-KEM	n=(1024 to 3072)	RSA-PKCS#1v1.5 (no CRT)	SP800-56B PKCS#1	Encryption, Decryption
SHA-2		Digest length: 224, 256, 384, 512	NIST FIPS 180-4	Message Digest
SHA-3		Digest length: 224, 256, 384, 512	NIST FIPS 202	Message Digest
Dilithium	ML-DSA-44 ML-DSA-65 ML-DSA-87		NIST FIPS 204 (Draft)	Key Generation Signature Verification Signature Generation
Kyber	ML-KEM-512 ML-KEM-768 ML-KEM-1024		NIST FIPS 203 (Draft)	Cryptographic Key Generation Key Encapsulation Key Decapsulation

LMS	LMS_SHA256_N32_H5(5)
	LMS_SHA256_N32_H10(6)
	LMS_SHA256_N32_H15(7)
	LMS_SHA256_N32_H20(8)
			SP800-208	Signature Verification

Algorithms	Key lengths (bits)	Modes	Specifications	Usage
	LMS_SHA256_N32_H25(9) LMOTS_SHA256_N32_W1 LMOTS_SHA256_N32_W2 LMOTS_SHA256_N32_W4 LMOTS_SHA256_N32_W8			

HSS	LMS_SHA256_N32_H5(5)
	LMS_SHA256_N32_H10(6)
	LMS_SHA256_N32_H15(7)
	LMS_SHA256_N32_H20(8)
	LMS_SHA256_N32_H25(9)
	LMOTS_SHA256_N32_W1
	LMOTS_SHA256_N32_W2
	LMOTS_SHA256_N32_W4
	LMOTS_SHA256_N32_W8
			SP800-208	Signature Verification

XMSS	XMSS-SHA2_10_256(1) XMSS-SHA2_16_256 (2) XMSS-SHA2_20_256 (3)		SP800-208	Signature Verification

XMSS^{MT}	XMSSMT-SHA2_20_2_256(1)
	XMSSMT-SHA2_20_4_256(2)
	XMSSMT-SHA2_40_2_256(3)
	XMSSMT-SHA2_40_4_256(4)
	XMSSMT-SHA2_40_8_256(5)
	XMSSMT-SHA2_60_3_256(6)
	XMSSMT-SHA2_60_6_256(7)
	XMSSMT-SHA2_60_12_256(8)
			SP800-208	Signature Verification

Table 4 Crypto Operation

3.3.4 Cryptographic Random Number Generation

The platform provides the application with a way based on DRBG to generate random numbers to as specified in NIST SP800-90A/B/C.

Conformance rationale:

The TRNG is compliant with FIPS-140. The ENT(P) utilizes FROs to supply entropy needed to generate random numbers. The ENT(P) meets the [SP800-90B] requirements.

3.3.5 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in algorithms listed in Table 5, as specified in Table 5, for key lengths specified in Table 5.

Algorithms	Key lengths (bits)	Modes	Specifications	Usage
CKG	128, 192, 256	AES key (modes ECB, CBC, CTR, CFB128, GMAC, CMAC, KWP)	SP800-133Rev2	Cryptographic Key Generation
	112 – 512 bit	HMAC key	SP800-133Rev2	Cryptographic Key Generation
	2048, 3072, 4096	RSA key pair	SP800-133Rev2	Cryptographic Key Generation

	P-224, P-256, P-384, P-521	ECDSA/EC Diffie-Hellman key pair	SP800-133Rev2	Cryptographic Key Generation
Dilithium	ML-DSA-44 ML-DSA-65 ML-DSA-87		NIST FIPS 204 (Draft)	Key Generation Signature Verification Signature Generation
Kyber	ML-KEM-512 ML-KEM-768 ML-KEM-1024		NIST FIPS 203 (Draft)	Cryptographic Key Generation Key Encapsulation Key Decapsulation

Table 5 Cryptographic Key Generation Description

Conformance rationale:

The TOE supports AES key generation via service “Generate Symmetric Key” through the supervisor application. See [API] 7.6.6

The TOE supports HMAC key generation via service “Generate Symmetric Key”. See [API] 7.6.6.

The TOE supports RSA keypair generation via service “Generate RSA Keypair”. See [API] 7.6.10.

The TOE supports ECDSA and ECDH keypair generation via service “Generate EC Keypair”. See [API] 7.6.9.

The TOE supports Dilithium Keypair generation via service “Generate Dilithium Keypair”. See [API] 7.6.39.

The TOE supports Kyber KEM Keypair generation via service “Generate Kyber KEM Keypair”. See [API] 7.6.42.

3.3.6 Cryptographic Keystore

The platform provides the application with a way to store cryptographic keys *listed in* Table 6 such that not even the application can compromise the *authenticity, integrity, confidentiality* of this data. This data can be used for the cryptographic operations *listed in* **Table** 4*.*

Key names	Location	Description	Cryptographic Operation
Netlist keysplits	RTL	Netlist keysplits are embedded into netlist as constants during synthesis and are common for all devices. Examples of these key splits are PNAK (Perso Netlist Keysplit), BNAK (Builder Netlist Keysplit), SNAK (SOC Netlist Keysplit).	Key derivation
DGOK (Device Generated OTP Keysplit)	OTP	DGOK is used to generate device-unique keys for certain operations.	Key derivation
General purpose keysplits written to OTP during device personalization	OTP	General purpose keysplits that can be programmed into OTP memory by the customer. These are used for generation of other keys.	Key derivation
Application specific keys	OTP, SRAM, external NVM	Application specific keys that are generated by the application.	Any cryptographic algorithms listed in **Table** 4

Table 6 Cryptographic Keystore

Conformance rationale:

There are 2 types of secure assets, dynamic and static. Dynamic assets are stored in SRAMCM (CMRT internal SRAM). Static assets are stored in OTP. See [API] 7.4 for details.

Confidentiality: Keysplits and derived keys are not readable by the CPU. Keys that are sent outside of CMRT should be wrapped and never exported in plaintext. (see [API] 7.4.5)

Authenticity: Access to OTP is protected by OMC (OTP Management Controller) managed permissions. Only m-mode can program OMC permissions.

Integrity: OTP is protected by ECC algorithm. See [HW_Manual] section 3.2.13.1.

3.4 Additional Security Functional Requirements

3.4.1 Secure Communication Support

The platform provides the application with a secure communication channel.

Conformance rationale:

The security of the communication between the secure processing environment and the CMRT as trusted subsystem is ensured by the proper physical integration of CMRT, as required per security objective HW_INTEGRATION. All communication between the secure processing environment and CMRT occurs over either an AXI or AHB bus which are not directly accessible nor influenceable externally. The channel is hardware implementation which ensures the integrity of the message.

Before establishing the session for executing any services, user should be authenticated via ECDSA challenge response. See [API] section 7.6.2. Anonymous users are not allowed.

3.4.2 Secure Communication Support – JTAG

The platform provides the application with a secure communication channel for debugging.

The channel authenticates a signed container with correct permission and for secure debugging.

Conformance rationale:

The JTAG interface is only used for secure debugging when the TOE is in the following lifecycle mode: Blank, Tested, Provisioned. After provisioning is finished, and the TOE is in Mission lifecycle mode, this interface is disconnected. In addition, the switching of TOE lifecycle requires authentication.

3.4.3 Secure Communication Support – KTI (Key Transport Interface)

The platform provides the application with a secure communication channel.

Conformance rationale:

The Key Transport Core (KTC) provides the interface for exporting keys from the TOE to the SoC over the Key Transport Interface (KTI). Before transferring of the keys, the user must first be authenticated via ECDSA challenge response. Only authenticated Crypto Officer or users are able to use this interface. The interface is only used for transferring keys securely. See [HW_Manual] 3.2.12 and [API] 7.6.13.

3.4.4 Secure Communication Enforcement

The Platform ensures the application can only communicate with the trusted subsystem over a secure communication channel.

Conformance rationale:

The security of the communication between the secure processing environment and the TOE as trusted subsystem is ensured by the proper physical integration of the TOE, as required per security objective HW_INTEGRATION. All communication between the secure processing environment and CMRT occurs over either an AXI or AHB bus and JTAG interface which are not directly accessible nor influenceable externally. AXI/AHB interface is used for processing commands and reply, as the main interface for communication. JTAG is only used for supporting secure debug functionality by sending authenticated commands. KTI is only used for securely transferring keys. Besides debugging and transferring keys, AXI/AHB interface is the only way of sending commands to the TOE. No other, potentially non-secure, communication interface exists.

3.5 Optional Security Functional Requirement

3.5.1 Secure External Storage

The platform ensures that all data stored outside the direct control of the platform, except for public data, is protected such that the confidentiality and integrity is ensured.

Conformance rationale:

Data can be sent outside of the TOE for purpose of persistent storage. All data is encrypted inside CMRT before sending outside, utilizing either Key Wrap With Padding (KWP) method as specified in SP800-38F or export via Key Transport Core (KTC) bus. See details: [API] section 7.6.12 and 7.6.13.

3.5.2 Secure debugging

The platform only provides a signed Containers with permissions to write to the TDV (Test Debug Vector) as specified in [HW_Manual] with debug functionality.

The platform ensures that all data stored by the application is made unavailable.

Conformance rationale:

The debug service is only available for signed Containers with permissions to write to the TDV. The TDV value is tied to the lifecycle of the TOE. Secure debugging is only allowed in lifecycle Blank, Tested and Provisioned. The debug interface is disconnected when the lifecycle is in Mission, RMA or Decommissioned. See [HW_Manual] section 7.2.

3.6 Product Life Cycle

3.6.1 Field Return of Platform

The platform can be returned to the vendor without user data.

Conformance rationale:

All security related data can be deleted. See details in [API] 7.6.34. Depending on the input of the value, the crypto officer can choose to delete different types of assets. In the case of decommission, all assets are zeroized, including:

- The root associated with the supervisor application
- The Crypto Officer’s root
- All valid User roots
- Dynamic assets
- All critical security parameters and keys

3.7 Compliance Functionality

3.7.1 Residual Information Purging

The platform ensures that *temporary data that will not be used anymore,* with the exception of data that will be used later, is erased automatically before the memory is used by the platform or application again and before an attacker can access it.

Conformance rationale:

The TOE automatically deletes temporary or intermediate data from the memory after performing a function. This ensures that all temporary data is deleted and cannot be accessed by an attacker.

3.8 Access Control

3.8.1 Authenticated Access Control

The platform allows Crypto Officer identified, authenticated, and authorized to allow performing all security functionalities provided by the platform.

The platform allows six users identified, authenticated, and authorized to allow performing all functionalities except the following:

- Create User
- Delete User
- Zeroize
- DRBG

Conformance rationale:

The TOE supports authentication of crypto officer and six user roles. Authentication must be firstly performed before executing any commands. Anonymous users are not supported. See [API] 7.6 for supported services. The user authentication is done via ECDSA challenge response.

4 Mapping and sufficiency rationales

This ST and associated TOE provide exact conformance to SESIP Profile for PSA Certified RoT Component Level 2.

4.1 Assurance

Assurance Class	Assurance Families	Covered by	Rationale
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	Section "Introduction" and "Title"	The ST reference is in the Title, the TOE reference in the "Platform reference", the TOE overview and description in "Platform functional overview and description".
	ASE_OBJ.1 Security requirements for the operational environment	Section "Security Objectives for the operational environment"	The objectives for the operational environment in "Security Objectives for the operational environment" refers to the guidance documents.
	ASE_REQ.3 Listed Security requirements	Section "Security requirements and implementation"	All SFRs in this ST are taken from [SESIP]. "Verification of Platform Identity" is included. "Secure Update of Platform" is included.
	ASE_TSS.1 TOE Summary Specification	Section "Security requirements and implementation"	All SFRs are listed per definition, and for each SFR the implementation and verification is defined in Security requirements and implementation
ADV: Development	ADV_FSP.4	Document [SYS_ARC], [API] used to meet this requirement	Complete set of TSF interfaces are well described
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	[SYS_ARC] [HW_Manual] [SEC_GUID]	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
	AGD_PRE.1 Preparative procedures	[HW_INT] [SEC_GUID]	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.

Assurance Class	Assurance Families	Covered by	Rationale
ALC: Life-cycle support	ALC_FLR.2 Flaw reporting procedures	Section “3.1.1”	The flaw reporting and remediation procedure is described.
ATE: Tests	ATE_IND.1 Independent testing: conformance	Functional testing as specified in document [ATE] and additional evaluator testing	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
AVA: Vulnerability Assessment	AVA_VAN.2 Vulnerability analysis	Vulnerability assessment is performed by the evaluator	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.

Table 7 Assurance

4.2 PSA Security Functions Mapping

PSA Security Function	Covered by SESIP SFR	Remark
F.INITIALIZATION	Secure Initialization	Full coverage by the BootFW and GPFW
F.SOFTWARE_ ISOLATION	Software Attacker Resistance: Isolation of Platform	Full coverage by isolating itself with other parts of the SoC.
	~~Software Attacker Resistance: Isolation of Application Parts~~	Not claimed
F.SECURE_ STORAGE	~~Secure Encrypted Storage~~	Not claimed
	~~Secure Storage~~	Not claimed
	~~Secure Encrypted Storage~~	Not claimed
	Secure External Storage	Stored data are encrypted.
F.FIRMWARE_ UPDATE	Secure Update of Platform	Full coverage by the secure booting mechanism (Fboot and Sboot)
F.SECURE_STATE	Software Attacker Resistance: Isolation of Platform	Full coverage by isolating itself with other parts of the SoC.
	Secure Initialization	Full coverage by the BootFW and GPFW
	Secure Update of Platform	Full coverage by the secure booting mechanism (Fboot and Sboot)
F.CRYPTO	Cryptographic Operation	Provides cryptographic algorithms
	Cryptographic KeyStore	Keys are securely stored in OTP and RAM
	Cryptographic Random Number	Provides NIST compliant TRNG
	Cryptographic Key Generation	Keys are securely generated

PSA Security Function	Covered by SESIP SFR	Remark
F.ATTESTATION	Verification of Platform Identity	Provides guidance on how to check system information.
	~~Verification of Platform Instance Identity~~	Not claimed
	~~Attestation of Platform Genuineness~~	Not claimed
	~~Attestation of Platform State~~	Not claimed
F.AUDIT	~~Audit Log Generation and Storage~~	Not claimed
F.DEBUG	Secure Debugging	Provides authenticated secure debugging service during provisioning stage
Additional security functionality (section 3.4 & 3.5)	Secure Communication Support	Only AXI or AHB bus could be configured to be used for communication.
	Secure Communication Enforcement	Only AXI or AHB bus could be configured to be used for communication.
	Field Return of Platform	Support deleting of all user data
	Residual Information Purging	The TOE automatically deletes temporary or intermediate data from the memory after performing a function.

Table 8 PSA Security Functions Mapping