QPG6200, with Secure Element

SESIP LEVEL 2 SECURITY TARGET

Revision History

Revision	Date	Description
0.1	*28/11/2023*	*Initial Draft.*
0.2	*20/12/2023*	*Updates made based on a review after client feedback.*
0.3	*16/12/2024*	*Platform scope includes modes of operation and Cryptographic Key Generation got refined*
0.4	*04/02/2025*	*Updates made based on an Observation Report*
0.5	*05/02/2025*	*Updates for the curve name, sefw version and editorial changes*
0.6	*05/03/2025*	*Minor editorial updates, mailbox command update, added variants information*
0.7	*08/04/2025*	*Updates made based on EM1*
0.8	*18/04/2025*	*Updates after SDK 1.0.1 Release*
1.0	*21/05/2025*	*Updates to the product naming*

Table of Contents

1 Introduction

This Security Target describes the core security features of the dedicated security management engine, the Secure Element (henceforth referred to as the platform) of the QPG6200 System-on-Chip. The evaluation gives the level of assurance provided by [SESIP] requirements (in chapter 2 "***Security requirements and implementation***") that a potential consumer can rely upon the product upholding, if they fulfil the objectives for the environment (in chapter 1.7 "***Security Objectives for the operational environment***"), in line with the requirements given by [Profile].

Note: The Security Target is based on [SESIP] methodology, version "Public Release".

1.1 ST Reference

This Security Target reference refers to the following:

Reference	**Value**
ST Name	*QPG6200* with *Secure element, SESIP Security Target*
ST Version	*Version 1.0*
ST Date	*May 21st, 2025*

Table 1 ST Reference

1.2 SESIP Profile Reference and Conformance Claims

This Security Target claims conformance to the following SESIP [Profile]:

Reference	**Value**
SP Name	*SESIP Profile for PSA Certified Level 2*
SP Version	*Version 2.0 REL*
Assurance Claim	*SESIP Assurance Level 2*
SESIP Standard	*[SESIP]*
Optional and Additional SFRs	*Base profile with optional Secure Debugging SFR*

Table 2 SESIP Profile for PSA Certified Level 2 Conformance Claims

1.3 Platform Reference

This Security Target claims conformance for the following configuration of the platform:

Reference	Value
Platform name	*Secure Element*
Platform version	Secure Element *ROM Bootloader v1.1* Secure Element FW SEUCFW-4.0.6-inc2
Platform identification	*QPG6200, with Secure Element*
Platform Type	*Secure subsystem on SoC*

Table 3 Platform Reference

1.4 Included guidance documents

The following documents are included with the platform:

Reference	Name	Version
[ST]	*QORVO SESIP Security Target*	*V1.0*
[2]	*Installation guide & User Manual*	*V1.0.1*

Table 4 Included Guidance Documentation

The following repository is provided online with the QPG6200 IoT Software Development Kit documents, it can be downloaded from the following URL:

Reference	Name	Information
[3]	Qpg6200-iot-sdk	https://gitlab.com/qorvo_sdk/public/devkits/qpg6200-iot-sdk.git
[3]	git@gitlab.com:qorvo_sdk/public/devkits/qpg6200-iot-sdk.git	2ca44aea3fa39ec52f1ee8f38c0982e32d708db0

Table 5 QORVO_SDK Repository Information

1.5 Lifecycle roles

The following user roles have been defined for the platform and are referenced in this document:

User Role	Description
System manufacturer	*The end user who purchases the product to develop it for resale.*
Chip manufacturer	*The manufacturer of the System-on-Chip is QORVO.*

Table 6 User Roles

1.6 Platform functional overview and description

The target of evaluation is the Secure Element subsystem and is referred as “SE”, “SEUC” or “platform” into the rest of this document. The term “application” refers to the rest of the SoC modules, which can invoke the SE features. The SE subsystem is integrated into the QPG6200 System-on-Chip (SoC) on which the applications are the APPUC domain. The current Security Target is covering the SESIP Profile for PSA Certified™ for the PSA Verified scheme.

1.6.1 Platform Security Features

The main security features of the platform are as follows:

- Secure unique identification
- Secure initialisation of the platform and SoC components
- Secure Updates
- Challenge interface for Debug
- Crypto Key storage and Operations
- Isolation of Secure Element

1.6.2 Platform Scope

The scope of the platform comprises of the Secure Element (SE) subsystem of the QPG6200.

As depicted below in Figure 1, components out-of-scope comprise the radio subsystem, the APPUC, the peripherals and the System Memories:

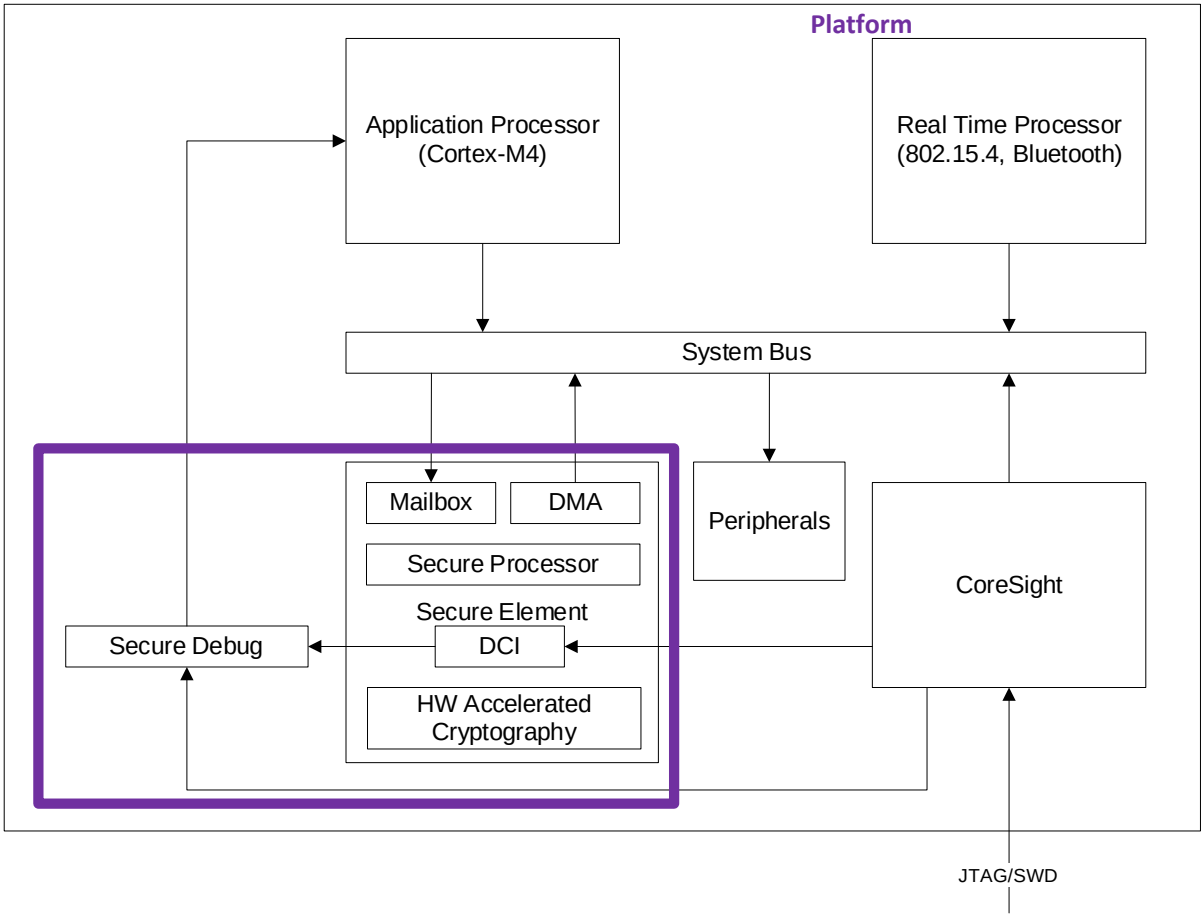

Figure 1 Scope of the Platform

The scope of the platform includes the following hardware components and interfaces:

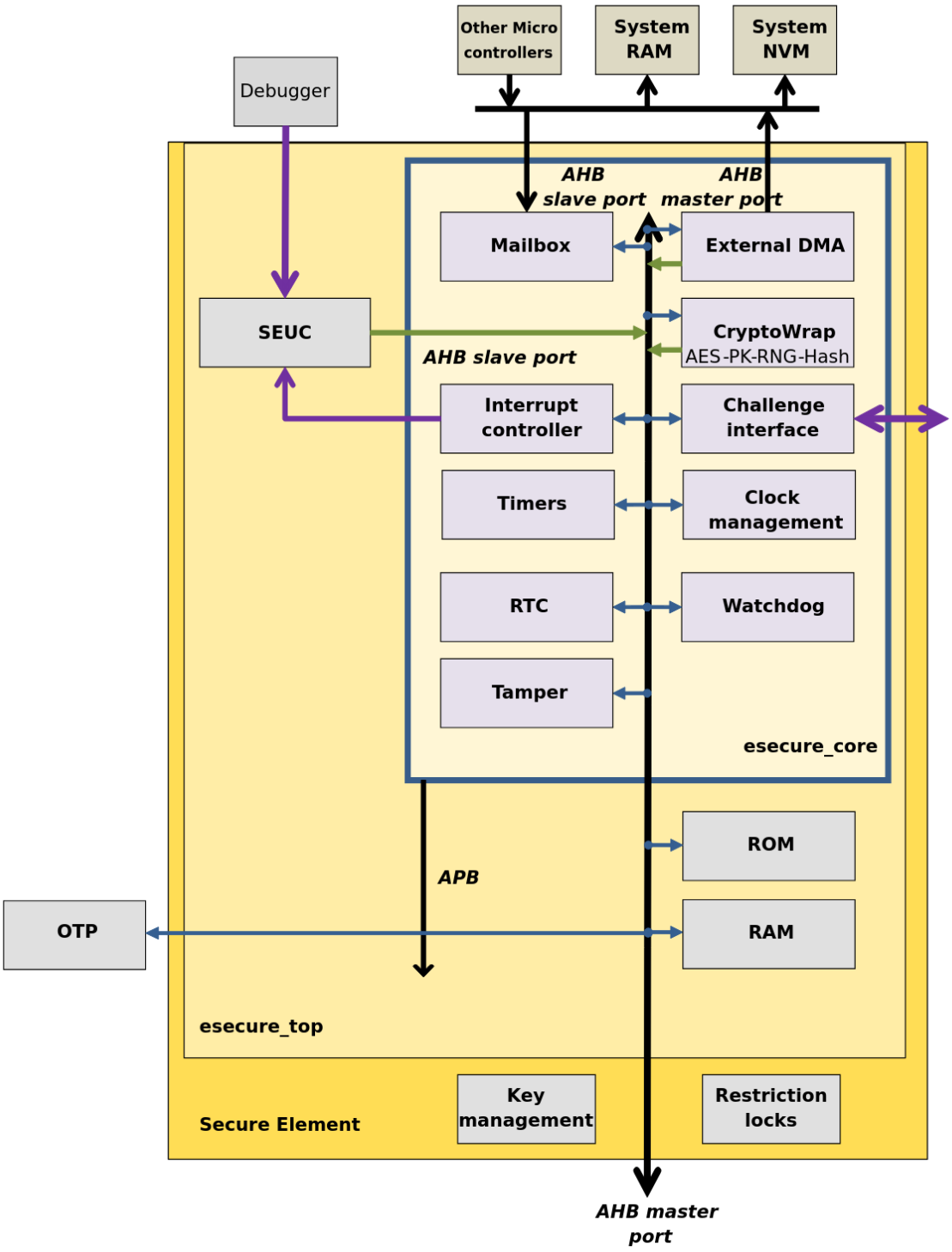

Figure 2. SE HW components

The scope of the platform includes the following software components and interfaces:

Component/Interface	Description
ROM Firmware	*Includes secure boot, secure update, life-cycle management, isolation features, and programmer and diagnostic mode management.*
SEFW	Includes key management, cryptographic operations, RNG and attestation features.
Mailbox	*IO interface with SE services from APPUC.*
DCI	*Debug Challenge Interface.*

Table 7 Software components and Interfaces

The block diagram below depicts the complete representation of the internal functioning of the platform:

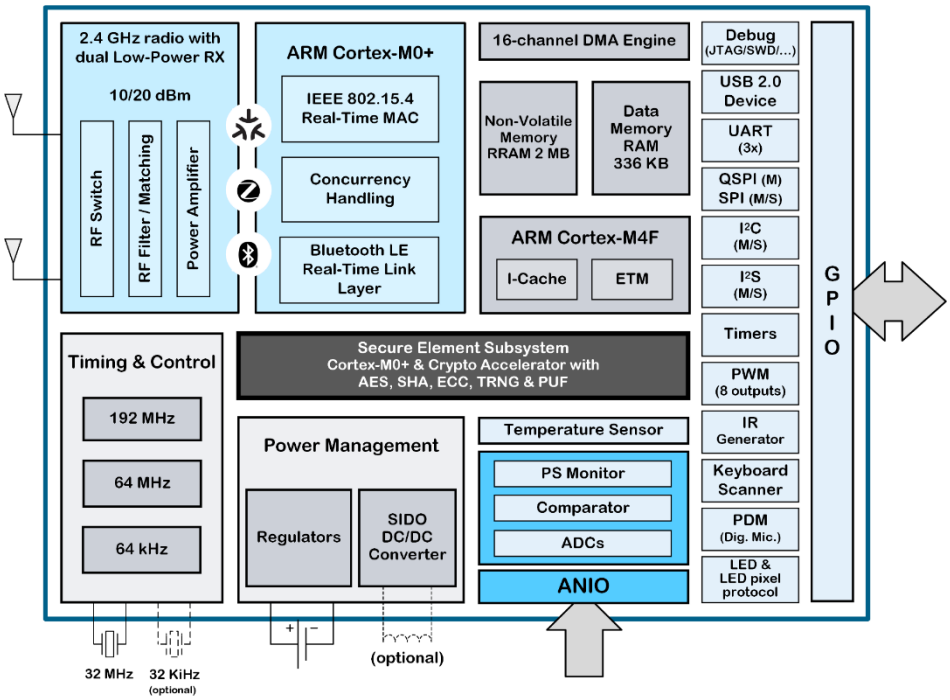

Figure 3 QPG6200 Block Diagram

1.6.3 Use Case Environments

The SE is part of the SoC which will be integrated into QPG6200, which requires a Hardware Root-of Trust to ensure the security of the final device use. QPG6200 is designed for use in RF Remote Control, Smart Home and other IoT applications. It consists of a radio transceiver, integrated real-time Medium Access Control and Bluetooth LE controller, integrated Arm Cortex-M4F microprocessor, RAM, NVM and OTP memory, security engine, event scheduler and an extensive set of peripherals including a configurable Keyboard Scanner and IR signal generator, thus enabling a single chip solution for remote control devices in the CE market.

The QPG6200 is intended to be purchased by a System Manufacturer for further development to be deployed into an IoT solution, combined with their own trusted user applications.

1.7 Security Objectives for the operational environment

For the platform to fulfil its security requirements, the operational environment (technical or procedural) must fulfil the following objectives:

Title	Description	Reference
TRUSTED_USERS	Actors in charge of Platform management, for instance for signature of firmware update, are trusted.	2.2.6 Secure Update of Platform
KEY_MANAGEMENT	Cryptographic keys and certificates outside of the Platform are subject to secure key management procedures.	*2.2.11* Cryptographic KeyStore
TRUSTED_ MANUFACTURERS	The System Manufacturer is trusted as it has the capability to use debug interfaces (no ability to modify SE-restricted memory areas).	*2.3.1* Secure Debugging
TRUST_PROVISIONING	Any secret to be provisioned into the platform is generated securely (e.g., via a standard compliant HSM) and subject to secure key management procedures. The provisioning process is done in secure sites with physical, logical security and organizational policies in place.	*2.2.3* Attestation of Platform Genuineness
PHYSICAL_PROTECTION	The SE lacks safeguards to protect it-self from physical attacks. Consequently, if there's a scenario where an adversary could potentially have physical access to the QPG6200, the customer must factor this into their risk analysis. They may need to incorporate mitigations at higher layers of the product or the overall system.	*The implementation of a root of trust which is inaccessible to an attacker and acts as the pillar of the verification on cascade of the device integrity assures that the device cannot be modified without notice. See* *Section 2.2.3* Attestation of Platform Genuineness *Section 2.2.4* Secure Initialization of Platform
RELIABLE_FLASH	The memory where the SE FW is stored is reliable and its access is controlled by the access control policies for restricted areas defined by the platform.	*2.2.7* Software Attacker Resistance: Isolation of Platform
UNIQUE_ID	The integrity and uniqueness of the unique identification of the platform must be provided by the platform user during the personalization stage.	*2.2.1* Verification of Platform Identity

1.7.1 Rationale for non-included Security Objectives

Title	Description	Rationale
UNIQUE_ID	The integrity and uniqueness of the unique identification of the platform must be provided by the platform user during the personalization stage.	*As the document [2] mentions in section Device Attestation, the assets which allow any*

		user to validate the device identity are incorporate by Qorvo. Since the platform user performs no action, the objective is not included in the scope of this ST.

2 Security requirements and implementation

2.1 Security Assurance Requirements

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

2.1.1 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:

QORVO defines and follows a strict PSIRP, implemented by their internal PSIRT team. As part of the PSIRP, which describes how to report a security incident to Qorvo, the actions taken by QORVO in response to such a report, what QORVO will do to address vulnerabilities and the way QORVO communicates relevant information regarding such reports and the actions taken to the relevant parties. This process provides an online publicly available interface (https://www.qorvo.com/support/product-security-incident-response-team) and includes the following four steps:

- **Report:** Vulnerabilities are reported to the PSIRT through an encrypted email (psirt@qorvo.com), the public key (PGP) is made available through the online interface. QORVO is committed to acknowledge the receipt of the report within 3 business days, and to provide regular updates throughout the handling process.
- **Assess:** The PSIRT will confirm the reported vulnerability and allocate it a unique Incident Report Number, the reporter will be informed of this by the PSIRT within 1 week of acknowledging the issue. During this phase, the PSIRT will estimate the Incident Security Level based on their assessment of the risk and its impact. The incident security level is a priority rating that will determine how the issue is handled. The PSIRT will then let the reporter know the Incident Security Level and the steps to be taken, along with an estimate of the timeline.
- **Solve:** The PSIRT will develop a solution to mitigate the reported vulnerability. Solutions will take different forms depending on the plan developed during the “Assess” phase. The PSIRT may work alongside other internal departments to develop the solution, though this is dependent on what is affected by the vulnerability. During this phase, progress updates will be provided to the reporter, additional information may be requested as well.
- **Communicate:** Disclosure of security incidents is at the discretion of the PSIRT and is only performed through private communication (encrypted email), this is an effort to limit exposure of the vulnerability. QORVO will take the necessary corrective actions to remediate the incident. Possible corrective actions are:

- Patch: release a patch that can be applied to the firmware of the platform.
- Release: include a fix in the next release of the affected firmware.

Note: Full detailed information on the PSIRT can be found in internal documents [1].

2.2 Base PP Security Functional Requirements

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

2.2.1 Verification of Platform Identity

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

Conformance rationale:

The platform contains dedicated memory areas used to retain identification information for the platform and its parts.

Identification of the parts of the platform can be retrieved by invoking the Mailbox interface; a secure element component that provides communication between the Customer Application and the Secure Element.

The SoC platform identification can be done using the APPUC accessible user infopage area. The user infopage starts at the NVM address 0x10200000.

The product id is 12 bytes in length. The first 10 bytes are an ASCII encoded string padded with 0. The string will start with QPG6200 followed by a single letter to identify the variant (with respect to packaging and power amplifier, J, L, M or N), The 11th byte is always 0x40, the 12th byte is always 0x00.

The product id can be read from the infopage base address with an offset of 0x00000180. Example product ID could be “515047363230304c00004000” which translates to “QPG6200L”.

The product version is of 4 bytes length, which can be read from the infopage base address with an offset of 0x0000018c. Example product version could be “01020400”. Product version should be decoded as x.y.z.u for each byte.

- x: 1 for QPG6200.
- y: indicates the version and will be 4 for the L variant, 5 for the other variants.
- z: Indicates the variant (packaging and power amplifier); 1 = J, 2 = L, 3 = M, 4 = N, 5 = P. See following table to see information about each variant.

Product Variant	Power amplifier	Package
QPG6200L	10dBm	4x4QFN32
QPG6200J	20dBm	4x4QFN32

QPG6200M	20dBm	5x5QFN40
QPG6200N	10dBm	5x5QFN40
QPG6200P	10dBm	4x4QFN32

Table 8 QPG6200 Variants Information

- u: reserved for future use, always 0

Commands are used to request a function be performed from the Mailbox. Version identification can be requested using a corresponding Mailbox command:

Reference	Mailbox Command	Mailbox Response
ROM BL Version SEUC Firmware	0x43230000	ROMBL 1.1 SEUCFW-4.0.6-inc2

Table 9 Verification of Platform Identity

2.2.2 Verification of Platform Instance Identity

The platform provides a unique identification of that specific instantiation of the platform, including all its parts.

Conformance rationale:

The serial number is a unique identification stored in eOTP, outside of the SE, it is composed by 128-bit string and is unique per device.

The platform instance serial number can be requested with mailbox command 0xFE000000.

2.2.3 Attestation of Platform Genuineness

The platform provides an attestation of the “Verification of Platform Identity” and “Verification of Platform Instance Identity”, in a way that ensures that the platform cannot be cloned or changed without detection.

Conformance rationale:

The platform retains a private key that is unique to each instance of the platform, this is known as the EK, the Endorsement Key, which is provided at production time. It is an ECDSA-P256 private key.

The following certificates are available on the device:

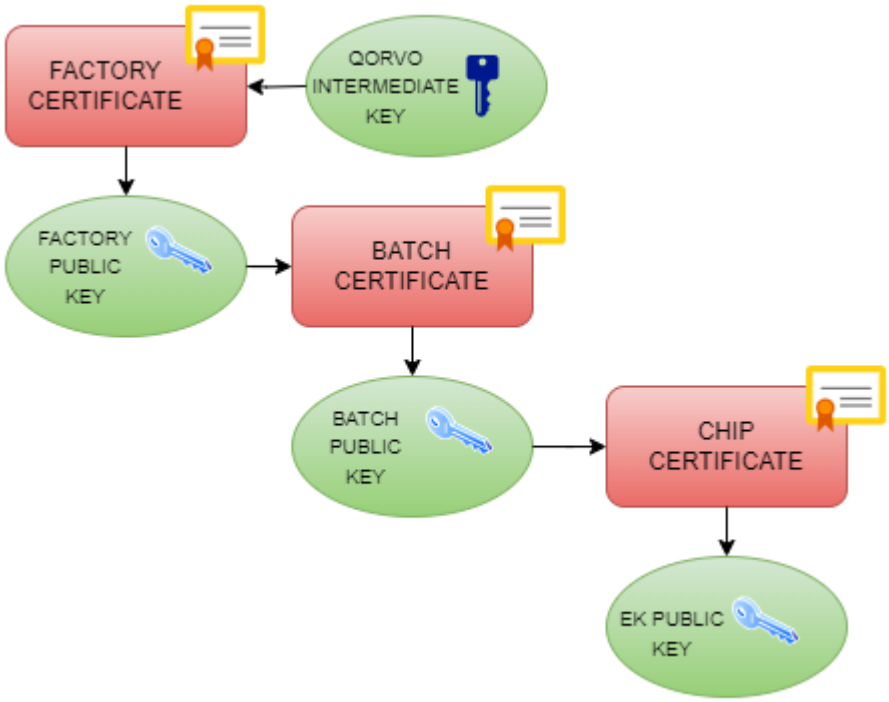

Figure 4 Certificates Chain

- The chip certificate is a certificate issued and signed by the chip manufacturer to identify the chip which contains the serial number of the chip and public part of the Endorsement Key (EK). This certificate is generated under-request and stored in volatile memory.
    - Mailbox command 0x0A000000

  The chip certificate format can be found in document [2].
- Batch certificate, stored as a binary in the eOTP, signs the batch public key with the factory private key
    - Mailbox command 0x43210000
- Factory certificate, stored in OTP, signs the factory public key with the Qorvo intermediate key
    - Mailbox command 0x43210100
- The intermediate certificate (signed by root key) and root certificate (self-signed) are made available on the Qorvo website for further verification.

2.2.4 Secure Initialization of Platform

The platform ensures its integrity and authenticity during platform initialization. If the platform integrity and authenticity cannot be ensured, the platform will go to a state where no other operation except optionally Secure Update of Platform can be performed.

Application Note:

The scope of this security target covers the Secure Element of the System-on-Chip, the remaining stages of the bootloader process are described in this requirement, but only for the purposes of providing clarity on the entire process. Thus, only the process for the Secure Element bootloader should be taken into consideration.

Conformance rationale:

The secure boot process ensures that only integer and authenticated SE FW runs on the platform:

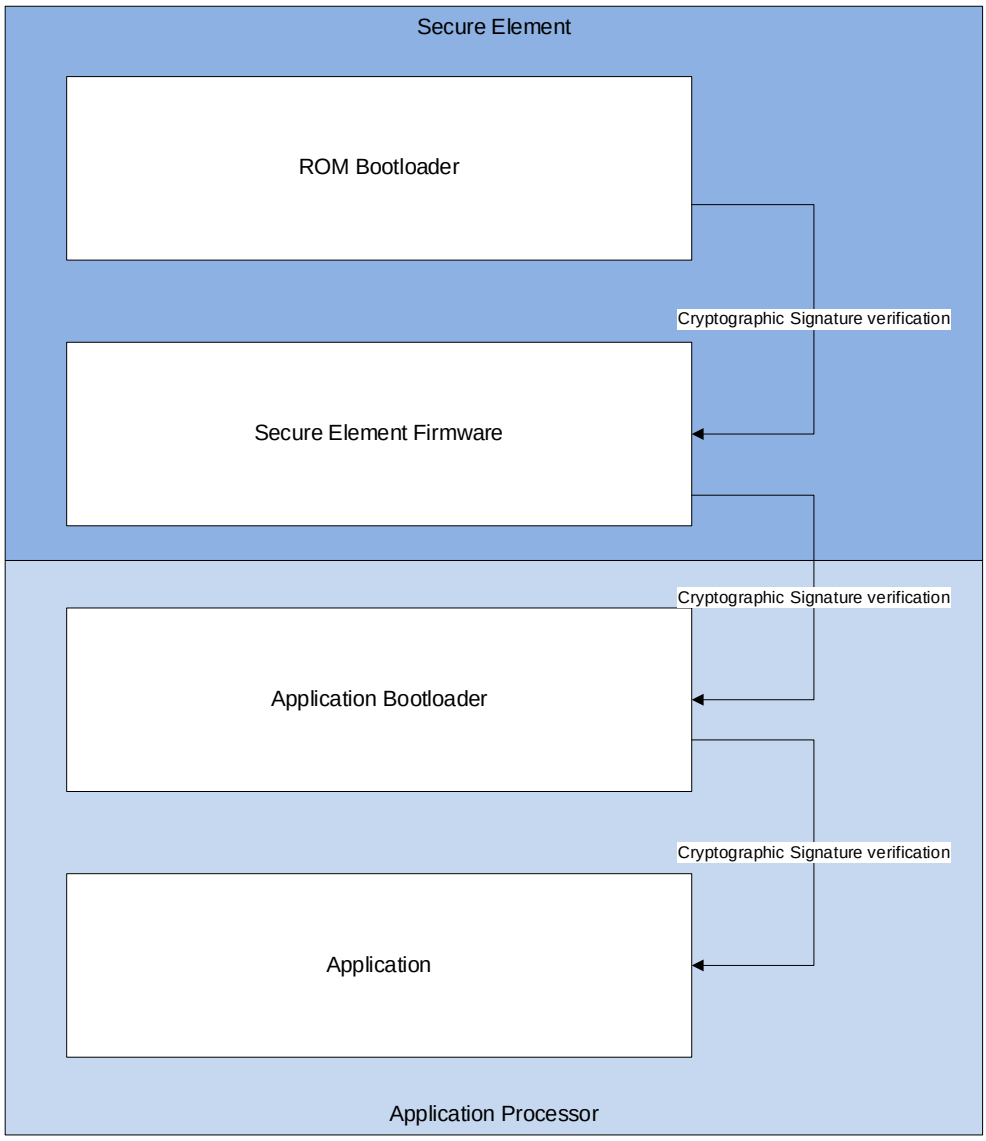

Figure 5 Secure Bootloader Process

When the device is powered up, the Application Processor and Real Time Processor are kept in reset. The ROM bootloader, being executed on the Secure Processor inside the Secure Element, initiates the Secure Boot process. The ROM bootloader provides an immutable Root of Trust in the hardware and verifies the SEFW ECDSA signature (secp256r1 with SHA256), including the signature over the Secure Element Firmware stored in RRAM. The verification is done against the hash of the certificate's public key stored under a controlled eOTP memory.

When the verification is successful, control of the Secure Element is transferred to the Secure Element Firmware (SEFW). Finally, the Secure Element Firmware verifies the ECDSA signature over the Application Bootloader.

In the event an error is detected during secure boot, after a finite number of recovery attempts, the ROM bootloader (ROMBL) enters a "diagnostic mode" served by the code

stored in Secure Element's ROM, which only includes basic non security-related commands in which the user can perform a secure update.

During the "diagnostic mode" it is possible to replace the SEUC firmware, which can be uploaded as an encrypted and signed upgrade blob. To protect the system, ROMBL will first decrypt and verify this blob prior to replacing the existing instance of the firmware.

2.2.5 Attestation of Platform State

The platform provides an attestation of the state of the platform, such that it can be determined that the platform is in a known state.

Conformance rationale:

The mailbox command 0x43230000 (get_platform_state) provides the following information:

- Current Life Cycle State
- SRK source
- Hash of ROMBL
- Hash of SEUCFW
- Version of ROMBL
- Version of SEUCFW

Refer to [2] for further information on the Mailbox command.

2.2.6 Secure Update of Platform

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

Application Note:

SW updates to the platform are made by updating the Secure Element firmware (SEFW) which is stored in RRAM. An updated version of the SEFW is delivered as an encrypted and signed binary image, comprised of the following:

- The full updated SEFW
- A rollback version (an integer between 0 and 63)

Conformance rationale:

QORVO may make available upgrades to the Secure Element Firmware. These upgrades come in the form of opaque binaries, encrypted with AES256 (AES-CTR) and cryptographically signed using ECDSA (secp256r1 with SHA256).

Once an upgrade binary has been downloaded to the platform, a Mailbox command is sent to the Secure Element to trigger the upgrade. This writes a magic word to a certain part of the locked piece of the RRAM. On reset, the ROMBL will look in the same area and if it contains the *magic word* it will try to find the upgrade blob, authenticate and write it in the secured part of the RRAM.

The updated firmware image is checked for its authenticity and integrity during the secure boot process, as described in section 2.2.4 “Secure Initialization of Platform”. A version number is associated with the firmware to prevent rollback.

The update mechanism for the SEUC FW is performed as follows:

- An update blob is placed within the application-writable section of RRAM:

  The implementation of the mechanism by which the application obtains the blob is out of scope and depends heavily on the customer application requirements.

  - The application may write the upgrade blob by means of the generic RRAM write mailbox command (0x43030000, WriteNVM)
  - It is also possible that a developer uploads the upgrade blob by means of the proprietary programmer interface or the JTAG/SWD debug interface.
- The “upgrade pending” flag is set, by one of the following methods:
  - The mailbox command 0x43120000 (StartUpgrade)
  - The challenge interface command 0x43120000 (StartUpgrade)
- The device is automatically reset
- The ROMBL will act on the “upgrade pending” flag by decrypting and verifying the upgrade blob:
  - If the verification fails, a finite number of retries will be tried before aborting the upgrade attempt

If the verification is successful, the decrypted image will be written over the active SEUC FW, and the device will reset to boot into the new SEUC FW.

Note: The SEUC FW signature is verified on each boot attempt by the process described in section 2.2.4. Refer to [2] for further information on Mailbox commands.

2.2.7 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.

Application Note:

The following [Profile] specific requirements are covered by this SFR:

- The PSA-RoT is isolated from the NSPE.
- The PSA-RoT is isolated from the Application Root of Trust Services.

Conformance rationale:

The Secure Element is a self-contained island containing a dedicated ARM Cortex-M0+ core. The application runs on a separate Cortex-M4F core (APPUC). A third separate core dedicated to remote communications is the radio subsystem (RTUC).

All cores have access to the RRAM; however, part of the RRAM is reserved for exclusive access to the SE, this reserved section stores the SEUC FW and other SE configuration and assets.

RTL logic in an external HW access controller controls the access to the restricted memory regions of the RRAM. The RTL description includes the logic that enforces access control based on the state of *lock bits*. During the secure boot, the RTL logic is designed to respond to specific events to initiate the configuration of lock bits accordingly. The ROM bootloader writes the value to the lock bits in the hardware access controller register corresponding to the SE reserved area.

At runtime, the hardware access controller continuously enforces the access control policies based on the lock bit configurations and the bus master ID requesting the operation.

The APPUC and RTUC must request security operations from the Secure Element, they cannot write to RRAM directly. As write operations are mediated by the Secure Element, they are requested through the mailbox interface.

2.2.8 Cryptographic Operation

The platform provides Operations in ***Table 10*** functionality with algorithms in ***Table 10*** as specified in specifications in ***Table 10*** for key lengths described in ***Table 10*** and modes described in ***Table 10***.

Algorithm	Operations	Specification	Key length [bits]	Modes
AES	Encryption, decryption, AEAD	NIST FIPS 197 (AES) NIST SP800-38A (ECB, CTR, CBC) NIST SP800-38C (CCM) NIST SP800-38D (GCM, GMAC) RFC 3610(CBC-MAC)	128, 192, 256	ECB, CTR, CBC, CFB CCM GCM, GMAC CBC-MAC

SHA	Secure hashing Keyed hashing for HMAC	FIPS 180-3	n/a	SHA-256 SHA-512 SHA-384
ECDSA	Signature generation Signature verification	NIST FIPS 186-5	Up to 521 bits	secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1,
ECDH	Key agreement	RFC 7748	Up to 521 bits	secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1
EdDSA	Signature generation Signature verification	RFC 8032	256	Curve25519 Ed448-Goldilocks
HMAC	Message authentication	FIPS 198-1	n/a	HMAC-SHA256 HMAC-SHA384
PBKDF2	Key derivation	NIST 800-132 FCD 18033-2	n/a	KDF2
HKDF	Key derivation	RFC 5869	n/a	HMAC-SHA256

Table 10. Cryptographic Operations

Conformance rationale:

The platform provides the aforementioned algorithms, key sizes and modes through HW crypto wrapper.

2.2.9 Cryptographic Random Number Generation

The platform provides a way based on a ***non-deterministic*** source to generate random numbers as specified in ***NIST-800-90B***.

Conformance rationale:

The True Random Number Generator provided by the platform uses a non-deterministic source for the generation of random numbers. The entropy is harvested from an "inverted gate oscillator" generating a continuous square wave output, the variations in the frequency of the square wave output are utilised as a source of randomness.

The NDRNG passes the NIST-800-22, AIS31 and NIST-800-90B test suites and has been used in several systems which are certified FIPS-140-2.

2.2.10 Cryptographic Key Generation

The platform provides a way to generate cryptographic keys for use in cryptographic algorithms in ***Table 11*** as specified in specifications in ***Table 11*** for key lengths described in ***Table 11***.

ID	Algorithm	Specification	Key Lengths
Symmetric	Symmetric	n/a	128, 192, 256 bits and any other value up to 512 bits.
ECC	Weierstrass Prime Curves	ISO/IEC 15946-1:2016	192
ECC	Montgomery Curves	RFC 7748	256 448
ECC	EdDSA	RFC 8032	256

Table 11.Cryptographic key generation

Conformance rationale:

The platform provides cryptographic key generation services for the application based on dedicated cryptographic resources not accessible from the CPU.

Generated keys are securely stored as described in section 2.2.11.

2.2.11 Cryptographic KeyStore

The platform provides a way to store ***cryptographic keys*** such that not even the application can compromise the ***authenticity***, ***integrity***, ***confidentiality*** of this data. This data can be used for the cryptographic operations: ***encryption***, ***decryption***, ***mac***, ***signature generation***.

Application Note:

The following Mailbox commands interact with the Cryptographic KeyStore:

- CreateKey
- ReadPubKey
- WrapKey
- UnwrapKey

Conformance rationale:

The platform provides secure cryptographic Key Storage to protect keys, and other sensitive data used by the application, by wrapping the data with a device-unique secret key; the Storage Root Key (SRK) which is stored in eOTP.

The SRK can be created in one of two possible ways:

- The result of a Physically Unclonable Function (PUF).

- Generated from the on-chip random number generator

For each item stored in Secure Storage, a unique key is derived from the Storage Root Key. This derived key is then used to encrypt the sensitive data (when storing keys, this is often referred to as 'wrapping' the key). The encryption or wrapping takes place in the Secure Element. The resultant encrypted or wrapped data can be stored in non-trusted memory.

Only the Secure Element can decrypt the data. Restrictions may be put on the usage of data in Secure Storage. For example, a key may be restricted to only be used as input to cryptographic operations. In this case, the key will never leave the Secure Element, protecting against accidental leakage.

Keys can be provisioned directly to Secure Storage, generated internally in the Secure Element or can be generated or negotiated by the Application.

Key wrapping is based on AES-GCM as described in document [2].

2.2.12 Secure Encrypted Storage

The platform ensures that all user data stored, except for ***none***, is encrypted as specified in ***NIST SP800-38D*** with a platform instance unique key of key length ***256 derived from the Storage Root Key***.

Conformance rationale:

The platform unique key is called storage root key, it is a unique AES per-platform key generated using either TRNG or PUF circuitry. This key is never used directly to wrap the data or keys stored by the secure storage. This is done by using derived keys that use the root storage key.

The Root Storage Key itself is stored in the OTP memory, accessible only to the SE. The appuc can wrap or unwrap the appuc data/key through the mailbox commands.

The wrapped data/key along with the tag is stored onto a volatile storage inside the secure element.

2.3 Optional Security Functional Requirements

The platform fulfils the following optional security functional requirements:

2.3.1 Secure Debugging

The platform only provides ***debug access port when debug is enabled*** authenticated as specified in ***[2]*** with debug functionality.

The platform ensures that all user data stored, with the exception of ***none***, is made unavailable.

Conformance rationale:

Access to QPG6200 debug interfaces is governed by the Secure Element, and can be configured by the System Manufacturer in one of two ways:

- Debug is enabled (protected by challenge interfaces)
- Debug is permanently closed

In case Secure Debug is configured, access to the debug features must be requested through the Debug Challenge Interface (DCI). After the GetChallenge DCI command (described in [2]) the SE expects the DebugAccess signed command. The key unblocking the interface belongs to the System Manufacturer.

The signature algorithm is ECDSA P-256 with SHA-256. A 2-level signature flow is used, which allows the manufacturer to delegate signing of debug commands and reduces usage of the top-level key.

With reference to Figure 3.1 of [2], the procedure is as follows:

- The SM generates a public/private key pair, the manufacturer's signing key and verifying key.
- The SM generates a public/private key pair, the developer's signing key and verifying key.
- The manufacturer signs the developer's verifying key, producing the developer's certificate.
- The developer uses their signing key to sign the command data.
- The developer sends the command data, signature and developer's certificate to the device.
- The device verifies the developer's certificate (using the manufacturer's public verifying key that was provisioned at production time)
- The device then verifies the command signature.

- If verification is successful, the requested debug mode will be applied.

Mapping and sufficiency rationales

This ST and associated platform provide exact conformance to Profile.

3.1 SESIP2 Sufficiency

Assurance Class	Assurance Families	Covered by	Rationale
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	Introduction	The “ST reference”, “Platform Reference” and “Platform Functional overview and Description” fulfils ASE_INT.1.
	ASE_OBJ.1 Security requirements for the operational environment	Introduction	The objectives for the operational environment in “Security Objectives for the operational environment” refer to the guidance documents.
	ASE_REQ.3 Listed Security requirements	Security requirements and implementation	All SFRs in this ST are taken from [SESIP].
	ASE_TSS.1 TOE Summary Specification	Security requirements and implementation	All SFRs are listed per definition, and for each SFR the implementation and verification are defined in Base PP Security Functional Requirements.
ADV: Development	ADV_FSP.4 Complete functional specification	Material provided to the evaluator	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	[2]	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
	AGD_PRE.1 Preparative procedures	[2]	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
ALC: Life-cycle support	ALC_FLR.2 Flaw reporting procedures	Flaw Reporting Procedure (ALC_FLR.2)	The flaw reporting and remediation procedure is described.

ATE: Tests	ATE_IND.1 Independent testing: conformance	Material provided to the evaluator.	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
AVA_VAN.2	AVA_VAN.3 Vulnerability survey	N.A. A vulnerability analysis is performed by the evaluator to ascertain the presence of potential vulnerabilities.	The platform evaluator will perform the vulnerability survey and evaluate the test results, assuming an attack potential of **Basic**.

Table 12 Assurance Mapping and Sufficiency Rationales

4 References

4.1 Evaluation Documents

[Profile] SESIP Profile for PSA Certified Level 2, PSA JSA JSADEN012, v2.0 REL 01

[SESIP] GlobalPlatform Technology, Security Evaluation Standard for IoT Platforms (SESIP), GP_FST_070, Public Release v1.2, July 2023

4.2 Developer Documents

[1] QORVO Product Security Incident Response Team, PSIRT, v0.2

[2] SDK v1.0.1 Installation Guidance

[3] QPG6200 IoT SDK Release 1.0.1

4.3 Terms and Definitions

API	Application Programming Interface
RoT	Root of Trust
SFR	Security Functional Requirement
PCB	Printed Circuit Board
EK	Endorsement Key
PUF	Physical Unclonable Function
APPUC	Application Microcontroller Subsystem
SEUC	Secure Element Microcontroller Subsystem
SE	Secure Element
PSIRT	Process Security Incident Response Team
PSIRP	Product Security Incident Response Process
Blob	Binary Large Object
SM	System Manufacturer
CM	Chip Manufacturer
NVM	Non-Volatile Memory
SEUC	Secure Element Microcontroller, Cortex-M0(+)
eOTP	One Time Programable