Security Target for the Renesas RA4M2 MCU Group

Based on [SESIP] methodology, version "Public Release v1.0"

Rev. 1.3
January 29, 2021

Contents

1 Introduction 2
1.1 ST reference 2
1.2 Platform reference 2
1.3 Included guidance documents 2
1.4 (Optional) Other Certification 3
1.5 Platform functional overview and description 3
2 Security Objectives for the operational environment 8
2.1 Platform Objectives for the Operational Environment 8
2.2 (Optional) Inherited Objectives for the Operational Environment 8
3 Security requirements and implementation 9
3.1 Security Assurance Requirements 9
3.1.1 Flaw Reporting Procedure (ALC_FLR.2) 9
3.1.2 Vulnerability Survey (AVA_VAN.1) 10
3.2 Security Functional Requirements 10
3.2.1 Verification of Platform Identity 11
~~3.2.2 Secure Update of Platform~~ 11
3.2.3 Verification of Platform Instance Identity 11
3.2.4 Decommission of Platform 11
3.2.5 Field Return of Platform 12
3.2.6 Limited Physical Attacker Resistance 12
3.2.7 Software Attacker Resistance: Isolation of Platform 13
3.2.8 Cryptographic Operation 13
3.2.9 Cryptographic Key Generation 14
3.2.10 Cryptographic KeyStore 15
3.2.11 Cryptographic Random Number Generation 15
4 Mapping and sufficiency rationales 17
4.1 SESIP1 sufficiency 17
5 References 19

Table of Figures

Figure 1 RA4M2 MCU Block Diagram 4
Figure 2 Flexible Software Package Block Diagram 5
Figure 3 Secure Crypto Engine 6
Figure 4 TOE Scope 7
Figure 5 True Random Number Generation 16

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 fulfill the objectives for the environment (in chapter “Security Objectives for the operational environment”).

The Security Target has been prepared for a SESIP Assurance Level 1 (SESIP1) evaluation.

1.1 ST reference

See title page.

1.2 Platform reference

TOE name	Renesas RA4M2 MCU Group
TOE version	1
TOE identification	RA4M2
TOE Type	General purpose microcontroller for connected applications

1.3 Included guidance documents

The following documents are included with the platform:

Reference	**Name**	**Version**
R01UH0892EJ0110	RA4M2 Group Hardware User’s Manual	Rev 1.10
R20UT4813EJ0100	Renesas Flash Programmer v3.08 Flash memory programming software User’s Manual	RFP v3.08, UM v1.00
R11AN0469EU0100	Renesas RA Family Device Lifecycle Management Key Installation Project	Rev 1.00
R11AN0468EU0100	Renesas RA Family Securing Data at Rest using Arm® TrustZone® Project	Rev 1.00
R11AN0467EU0100	Renesas RA Family Security Design with Arm TrustZone IP Protection Project	Rev 1.00
R20AN0577EG0101	Renesas RA Family Arm TrustZone Tooling Primer	Rev 1.01
R01AN5562EJ0100	Standard Boot Firmware for the RA family MCUs Based on	Rev 1.00

	Arm® Cortex®-M33	

1.4 (Optional) Other Certification

The product has previously been evaluated following [ARM PSA L1]:

Scheme	PSA Certified Level One
Certification body	TrustCB
Certification number	0716053549754-10010
Certificate date	25/02/2020

1.5 Platform functional overview and description

The flexible Renesas RA Family 32-bit MCUs are industry leading 32-bit MCUs with the Arm® Cortex®-M33, -M23 and -M4 processor cores and PSA certification. RA delivers key advantages compared to competitive Arm Cortex-M MCUs by providing stronger embedded security, superior CoreMark® performance, and ultra-low power operation. PSA certification provides customers the confidence and assurance to quickly deploy secure IoT endpoint and edge devices, and smart factory equipment for Industry 4.0.

The RA4M2 is a member of the new generation of security-focused RA Family MCUs from Renesas. State-of-the-art security features combined with best-in-class peripheral IP and feature- and pin-compatibility between the MCU Series make RA Family MCUs the optimum choice for nearly any connected embedded product.

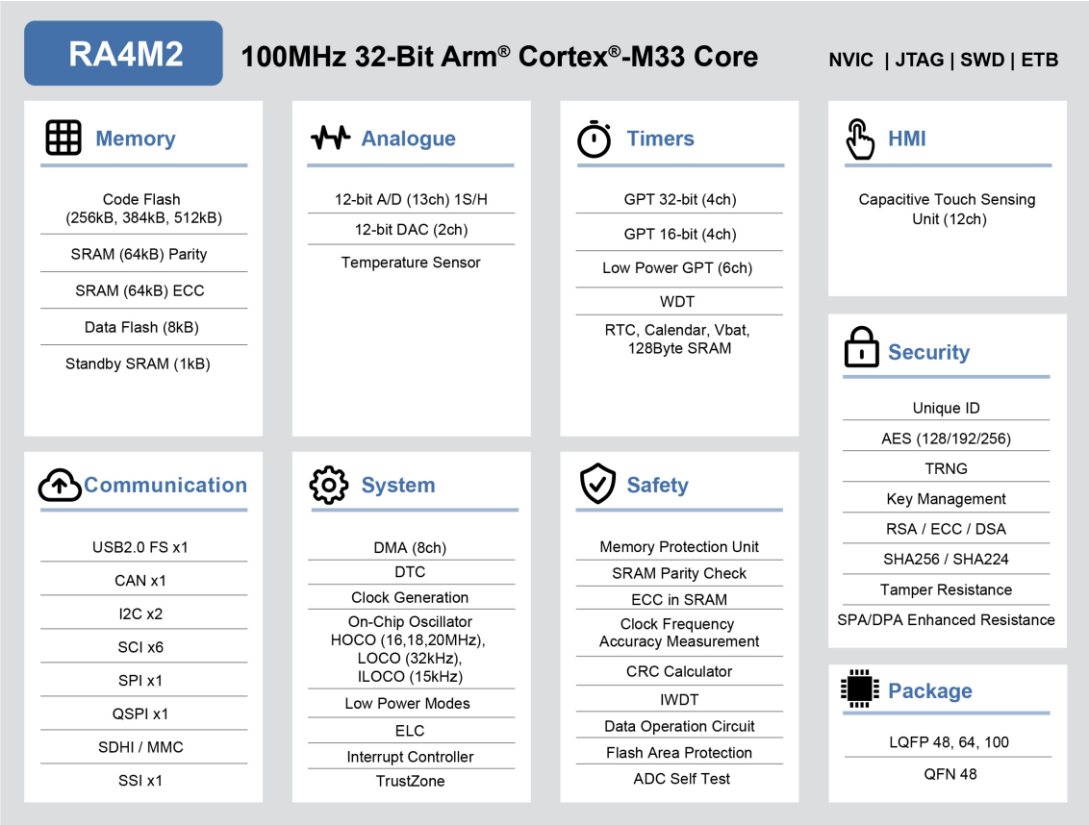

Figure 1 RA4M2 MCU Block Diagram

The RA Family Flexible Software Package (FSP) is an enhanced software package designed to provide easy-to-use, scalable, high-quality software for embedded system designs using Renesas RA family of Arm Microcontrollers. With the support of new Arm TrustZone and other advanced security features, FSP provides a quick and versatile way to build secure, connected IoT devices using production ready drivers, FreeRTOS™, and other middleware stacks.

FSP includes best-in-class HAL drivers with high performance and low memory footprint. Middleware stacks with FreeRTOS integration are included to ease implementation of complex modules like communication and security. The e² studio IDE provides support with intuitive configurators and intelligent code generation to make programming and debugging easier and faster.

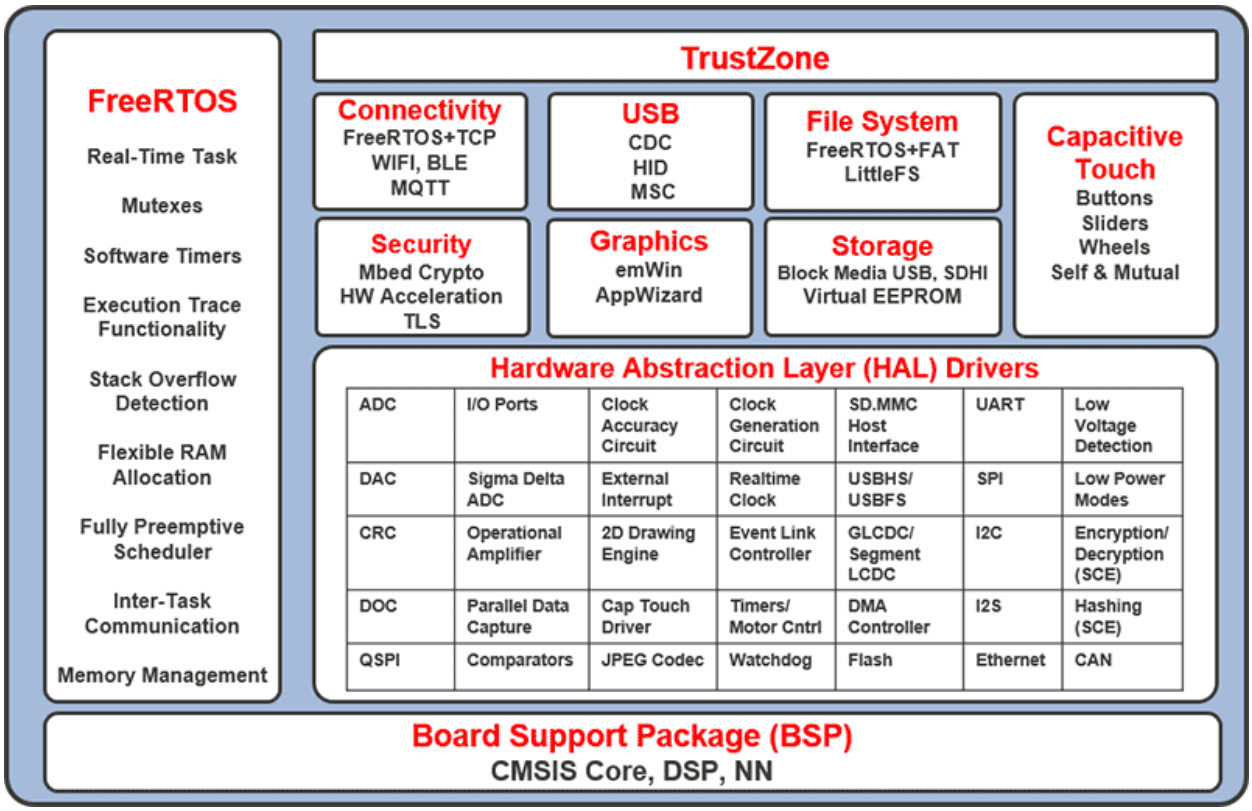

Figure 2 Flexible Software Package Block Diagram

The TOE consists of a microcontroller containing unique hardware security features designed to enable best-in-class security for a broad range of connected devices.

The TOE is intended to be used by product developers who need to protect valuable assets from a wide range of remote and physical attacks.

The main security features of the TOE are as follows:

- Arm Cortex-M33 32-bit core with Arm TrustZone® technology as the hardware-enforced isolation mechanism
- Integrated Secure Crypto Engine (SCE9) providing AES, RSA, ECC, and hash generation in an isolated on-chip environment
- Secure unlimited key storage utilizing a factory programmed 256-bit Hardware Unique Key (HUK)
- 128-bit unique identifier
- True Random Number Generator (TRNG)
- Advanced Device Lifecycle Management capability

The Secure Crypto Engine (SCE9) is a dedicated crypto subsystem contained within the MCU. Protected by an Access Management Circuit that can shut down the crypto engine upon illegal access attempts, the SCE9 performs all plaintext crypto operations using its own dedicated internal

RAM, which is not accessible by any CPU-accessible bus. The advanced key storage and key handling capabilities of the SCE9 can ensure that no plaintext keys are ever stored in CPU-accessible RAM, code flash, or external storage.

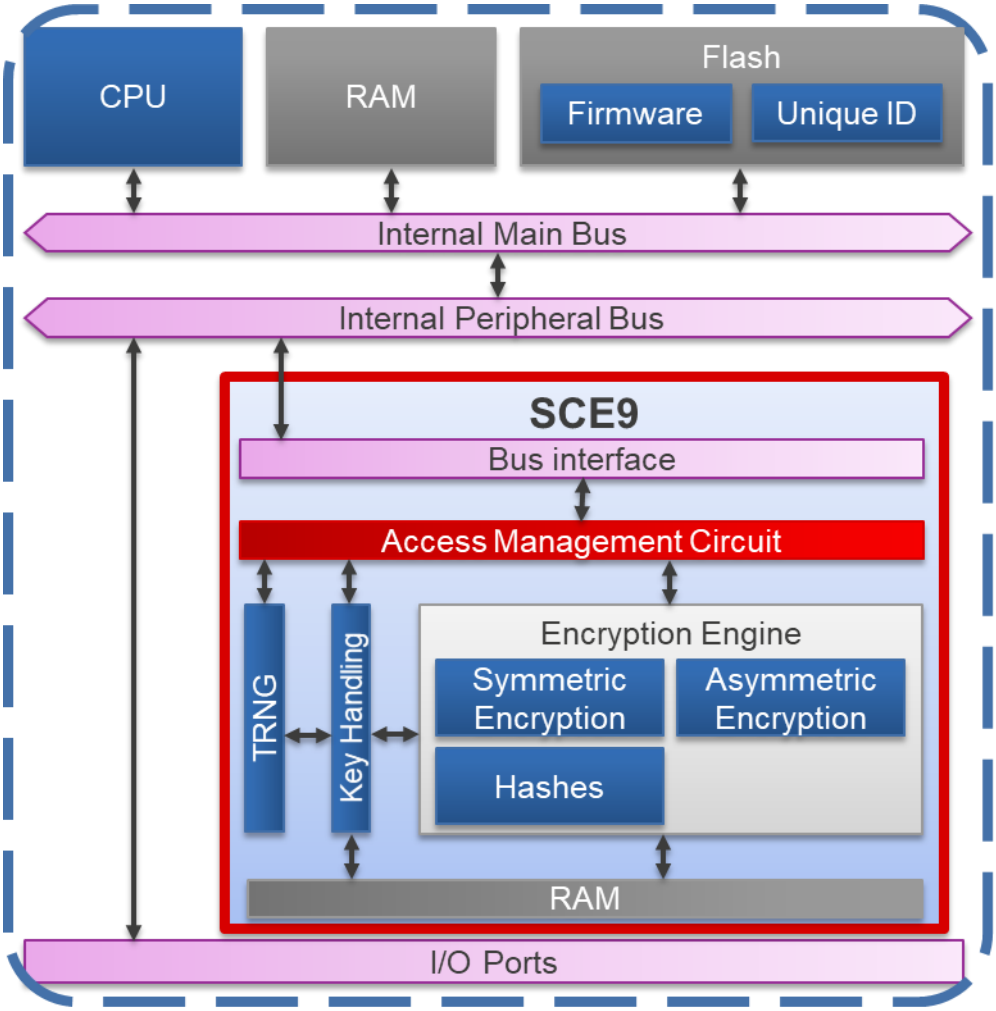

Figure 3 Secure Crypto Engine

The TOE scope is depicted in **Figure 4 TOE Scope**. The blue parts are within the evaluation scope and the gray parts are outside of the evaluated scope. The out of scope part comprises the Flexible Software Package (FSP) and any application software created by the end product developer.

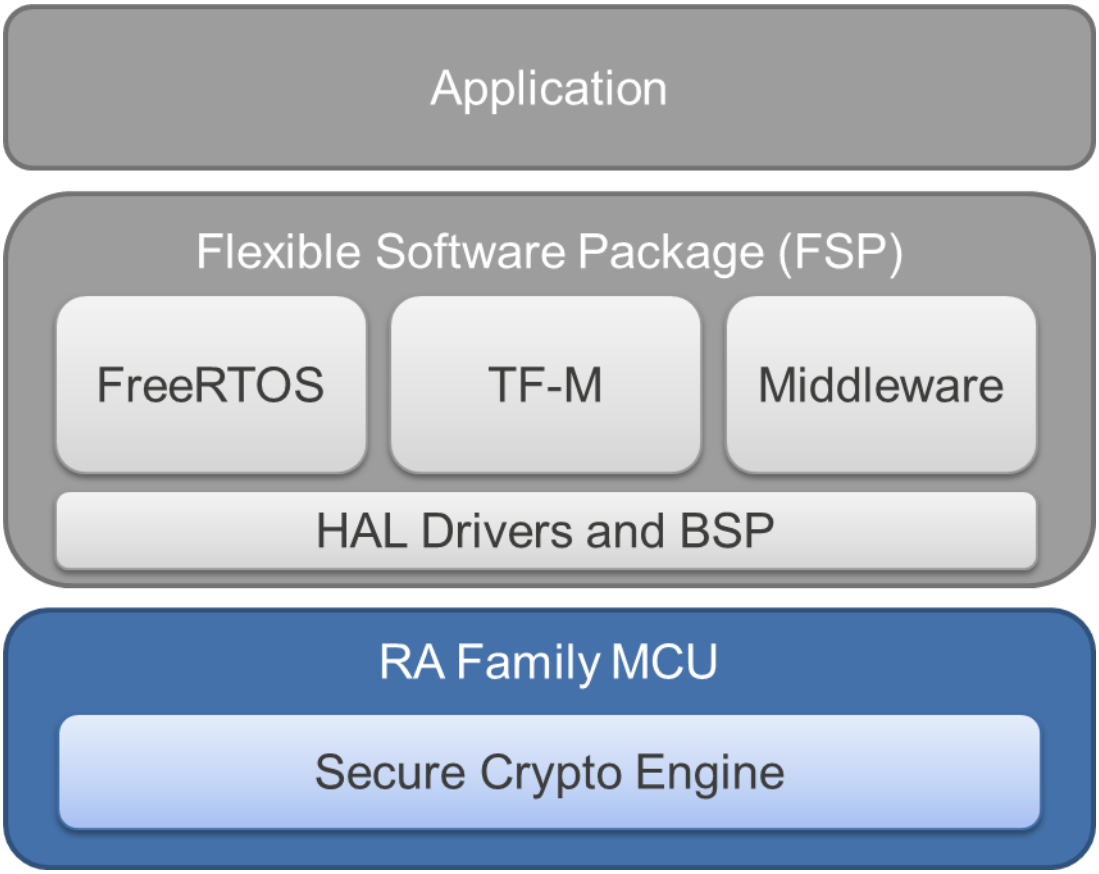

Figure 4 TOE Scope

The physical scope includes only the RA Family MCU itself, with the functional blocks identified in **Figure 1 RA4M2 MCU Block Diagram**. Guidance documentation for the MCU is listed in section **1.3 Included guidance documents** and is available for public download from the Renesas web site (www.renesas.com). RA Family MCUs are available via all of Renesas's normal sales channels. All RA Family MCUs contain the security features listed in this Security Target; no special part number nor personalization is required.

The logical scope includes the hardware and software interfaces necessary for the application and platform software to utilize the hardware. Cryptographic functions, such as encryption/decryption, hashing, secure cryptographic key handling, and random number generation, are performed by the Secure Crypto Engine (SCE9), shown in **Figure 4 TOE Scope** and detailed in **Figure 3 Secure Crypto Engine**. The remaining security features, including immutable MCU identification information, isolation mechanism, and external interface disabling, are provided by other silicon features on the MCU. These features are described in detail in the MCU documentation listed in section **1.3 Included guidance documents**.

Platform and application software is not in scope of this evaluation. No non-TOE hardware/software/firmware is required.

2 Security Objectives for the operational environment

2.1 Platform Objectives for the Operational Environment

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

- The end product developer shall ensure that the MCU is configured in the appropriate DLM state for the specific stage of the product's life cycle, as described by section ***46.3 Device Lifecycle Management*** in the Hardware User Manual listed in section 1.3 above.

- The end product developer shall ensure that TrustZone and the device MPUs are configured correctly for the application's memory isolation requirements, as described in sections ***2.5 Security Attribution for Memory***, ***15 Memory Protection Unit (MPU)***, ***46.2 Arm TrustZone Security***, and ***46.5 Register Descriptions*** in the Hardware User Manual listed in section 1.3 above.

- The end product developer shall ensure that TrustZone and the device MPUs are configured correctly for the application's peripheral and pin isolation requirements, as described in sections ***15 Memory Protection Unit (MPU)***, ***46.2 Arm TrustZone Security***, and ***46.5 Register Descriptions*** in the Hardware User Manual listed in section 1.3 above.

- For optimal key protection, the end product developer shall install any externally-generated keys as per section ***46.4 Key Injection*** in the Hardware User Manual listed in section 1.3 above.

2.2 (Optional) Inherited Objectives for the Operational Environment

Not applicable.

3 Security requirements and implementation

3.1 Security Assurance Requirements

The claimed assurance requirements package is: **SESIP1** as defined in [SESIP].

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 give generate any needed update and distribute it, the developer has defined the following procedure:

Renesas has a dedicated team that is responsible for the overall management of monitoring, investigating, and communicating security issues. Renesas PSIRT (Product Security Incident Response Team) operates as an independent unit with assigned window persons for every Renesas business unit, to ensure that internally and externally identified security vulnerabilities are captured, communicated, and addressed across all Renesas groups and any affected customers.

The publicly available interface can be accessed via www.renesas.com/psirt, whereby individuals and/or companies outside Renesas can securely submit vulnerability reports through a dedicated email address (renesas_psirt@lm.renesas.com) using PGP encryption.

Once a vulnerability has been entered into the system, from either the public interface or internally, internal Renesas operating procedures for corrective action of security incidents and vulnerabilities govern the processing of the vulnerability. This process includes the following steps:

- Reporting the security issue – Security issues can be reported for PSIRT processing via the external web/email interface, from Renesas internal product design teams via the PSIRT window persons, or by PSIRT members directly. If the external report is received from a Renesas customer, Renesas QAD (Quality Assurance Division) will also be involved in the ensuing communications.
- Investigating the security issue – PSIRT confirms the existence, reproducibility, and threat assumptions in the report. Once confirmed, PSIRT works closely with the relevant product design team(s) to ensure that addressing the security issue is given appropriate priority, with consideration given to the scope of the affected product(s), the seriousness of the vulnerability, and the feasibility of an attack.
- Taking actions for the security issue – The product design team works with PSIRT to determine a corrective action plan. If the issue is software-related, updated software and user manual, security manual, and/or other equivalent documentation is created for distribution to customers. If the issue is silicon-related, the corrective action most likely will require a new product revision. Until the revision update is performed, usage restrictions and recommendations will be added to the user manual, security manual, and/or other equivalent documentation.
- Notifying the customer – The product design team creates a corrective action plan, which includes a plan of action, notification of interested parties both internal and external to Renesas, and completion schedule. Depending on the production status and origin of the vulnerability report, either the product design team or PSIRT will handle communication

with the report originator. For any issues in released products that affect Renesas customers, Renesas QAD will serve as the main point of contact to ensure that Renesas customers are informed of the vulnerability and can appropriately address the issue in their end products.

The silicon and built-in factory programming bootloader of the platform cannot be updated or patched. However, a secure boot solution can be implemented in firmware that can verify the integrity and authenticity of the code running on the platform as well as any application updates. The update mechanism must be implemented by the customer. Renesas provides a sample implementation and works with multiple third-party partners to enable their security solutions to run on the Renesas platform. Those mechanisms are not within scope of this evaluation.

3.1.2 Vulnerability Survey (AVA_VAN.1)

In accordance with the requirement for a vulnerability analysis survey (AVA_VAN.1) the developer has performed a vulnerability survey and submits the following test results to demonstrate the consideration of publicized potential vulnerabilities relating to the TOE:

The Secure Crypto Engine, isolation mechanism, programmer/debugger interface, and Device Lifecycle Management mechanism components of the platform were analysed to determine their vulnerability and attack potential. Side channel attacks, fault injection, abuse of the programmer/debugger interface, RNG attacks, and physical extraction of device information by decapping the device were considered and deemed to have an attack rating of 21 or greater. Attacks that are within scope for the SESIP1 assurance level are those attacks that have an attack rating of 0-15. Therefore, these attacks are out of scope for this evaluation.

Also, public databases were searched for publicly known vulnerabilities:

- Common Vulnerabilities and Exposures (CVE, https://cve.mitre.org)
- Common Weakness Enumeration (CWE, https://cwe.mitre.org)
- Common Attack Pattern Enumeration and Classification (CAPEC, http://capec.mitre.org/data/index.html)

No vulnerabilities were found.

The Arm Cortex-M33 core with TrustZone was also assessed for vulnerabilities, using the databases listed above. No vulnerabilities were found for the core itself, but CVE-2020-16273 stack sealing vulnerability has been reported against this hardware. This CVE is out of scope for this evaluation, as it pertains to software running on the TOE, not the TOE itself. Software implemented on this TOE should address this vulnerability, as per Arm's recommendation: https://developer.arm.com/support/arm-security-updates/armv8-m-stack-sealing.

3.2 Security Functional Requirements

The platform fulfills 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 contains a 16-byte read-only register that contains an ASCII representation of the device part number. The PNRn Part Numbering Registers are documented in section ***44.4.6 PNRn: Part Numbering Register*** in the Hardware User's Manual (R01UH0892EJ0110). A part number listing is given in section ***1.3 Part Numbering***.

The part number information is written as part of the production process, and the production testing procedures verify the value has been written correctly.

~~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.~~

The platform does not support the update or patching of hardware nor internal boot firmware, which is stored in unmapped memory that is not accessible by application code, a debugger, nor a device programmer. It does offer features that enable a customer to implement secure update mechanisms for the own code.

Justification for why this platform does not support secure updates is provided as part of section ***3.1.1 Flaw Reporting Procedure (ALC_FLR.2)*** in this document.

3.2.3 Verification of Platform Instance Identity

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

Conformance rationale:

The platform contains a 16-byte read-only register that contains a 16-byte ID code that is guaranteed to be unique for each MCU. The UIDRn Unique ID Registers are documented in section ***44.4.5 UIDRn: Unique ID Register*** in the Hardware User's Manual (R01UH0892EJ0110).

The unique ID is written as part of the production process, and the production testing procedures verify the value has been written correctly.

3.2.4 Decommission of Platform

The platform can be decommissioned.

Conformance rationale:

The Device Lifecycle Management states of the platform allow the MCU to be delivered to the end customer in what is referred to as the DPL (Deployed) state. In this state, debugging and reading/programming of the device are not available. Details of the platform's device lifecycle

management states and operation are documented in section ***46.3 Device Lifecycle Management*** in the Hardware User's Manual (R01UH0892EJ0110).

To decommission the platform with destruction of all data, complete erasure of the MCU is allowed via the serial programming interface. Note that if any flash blocks have been locked to be permanently immutable, this function is not available.

The Device Lifecycle Management mechanisms are thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.5 Field Return of Platform

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

Conformance rationale:

The Device Lifecycle Management states of the platform allow the MCU to be changed from either the SSD (Secure Software Development) or DPL (Deployed) state to the RMA_REQ (Return Material Authorisation Request) state in preparation for returning the device to Renesas for failure analysis. In this state, debugging and reading/programming of the device are not available. Details of the platform's device lifecycle management states and operation are documented in section ***46.3 Device Lifecycle Management*** in the Hardware User's Manual (R01UH0892EJ0110).

To enable the capability to perform this state transition, the end product developer must install an RMA authentication key RMA_KEY while the device is in the SSD (Secure Software Development) state. When this state transition is requested, the request will be authenticated using this key. Upon successful authentication and transition to the RMA_REQ state, all flash memory is erased except any flash blocks that have been permanently locked.

The Device Lifecycle Management mechanisms are thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.6 Limited Physical Attacker Resistance

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises ***Verification of Platform Identity, Verification of Platform Instance Identity, Software Attacker Resistance: Isolation of Platform, and/or Cryptographic KeyStore***.

Conformance rationale:

The platform prevents compromising the SFRs "Verification of Platform Identity" and "Verification of Platform Instance Identity" by placing these values in read-only flash. This prevents an attacker from being able to modify the platform or platform instance identity. These SFRs can be further protected by placing the device in the LCK_BOOT (Lock Boot Interface) DLM state, which will prevent an attacker from being able to read these values. See sections ***44.4.5 UIDRn: Unique ID Register***, ***44.4.6 PNRn: Part Numbering Register***, and ***46.3 Device Lifecycle Management*** in the Hardware User's Manual (R01UH0892EJ0110) for more information.

The platform prevents compromising the SFR "Software Attacker Resistance: Isolation of Platform" by enforcing TrustZone isolation through the use of an IDAU whose settings are programmed into

nonvolatile memory when the device lifecycle is in the SSD (Secure Software Development) state. These memory security attributions are loaded into the IDAU and the memory controller before application execution. They cannot be updated by an application, preventing an attacker from compromising the "Isolation of Platform" SFR. See section ***2.5 Security Attribution for Memory*** in the Hardware User's Manual (R01UH0892EJ0110) for more information.

The platform prevents compromising the SFRs "Cryptographic KeyStore" by having the capability to store all symmetric and asymmetric private keys in an MCU-uniquely wrapped format. Key wrapping guarantees the authenticity, integrity, and confidentiality of the key – if the wrapped key is altered, it becomes unusable on the MCU that wrapped it, and if it is copied to another MCU, that MCU cannot successfully unwrap it. The platform does not support unwrapping a wrapped key to expose the plaintext key. Wrapped keys stored on the MCU can be further protected by storing them in locked flash blocks to make them immutable and by placing the device in the DPL (Deployed), LCK_DEBUG (Lock Debugger), or LCK_BOOT (Lock Boot Interface) DLM state. See sections ***36 Secure Crypto Engine (SCE9)*** and ***46.3 Device Lifecycle Management*** in the Hardware User's Manual (R01UH0892EJ0110) for more information.

The part number and unique ID information is written as part of the production process, and the production testing procedures verify the value has been written correctly.

The referenced silicon functionality is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.7 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.

Conformance rationale:

The platform contains an Arm® Cortex®-M33 core based on the Armv8-M architecture with Arm® TrustZone® technology. TrustZone allows the application developer to isolate assets and services in a Trusted region, with Untrusted code only able to utilise those services that have been explicitly made available for use.

The isolation mechanism is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.8 Cryptographic Operation

The platform provides the application with ***the cryptographic operations from Table 1*** functionality with ***the specified algorithms listed in Table 1*** as specified in ***relevant specifications shown in Table 1*** for key lengths ***listed in Table 1*** and modes ***listed in Table 1***.

Table 1 Cryptographic Operations

Algorithm	Operations	Key Length	Modes	Specification
AES	Encryption, decryption	128, 192, 256 bits	ECB, CBC, CTR, CMAC, CCM, GCM, XTS, GCTR	NIST FIPS PUB 197 NIST SP800-38A (ECB, CBC, CTR) NIST SP800-38D (GCM) IETF RFC 3610 (CCM)
RSA	Signature generation, signature verification, public-key encryption, private-key decryption	1024, 2048, 3072,4096 bits	n/a	IETF RFC 8017 FIPS PUB 186-4
ECC	Signature generation, signature verification	Curves: NIST P-192, P-224, P-256, P-384; Brainpool P256r1, P384r1, P512r1	n/a	FIPS PUB 186-4
SHA2 Hash	SHA-224, SHA-256	n/a	n/a	FIPS PUB 180-4
HMAC	HMAC-SHA224, HMAC-SHA256 cryptographic hash	224, 256 bits	n/a	FIPS 198-1

Conformance rationale:

The above crypto operations are supported by the Secure Cryptographic Engine (SCE9), as documented in section ***36 Secure Cryptographic Engine (SCE9)*** in the Hardware User's Manual (R01UH0892EJ0110).

The Secure Cryptographic Engine is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.9 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in ***AES*** as specified in ***NIST FIPS PUB 197*** for key lengths ***128, 192, and 256 bits***.

Conformance rationale:

AES key generation is performed by creating a random number consisting of the required number of bits to act as the encryption/decryption key. The platform's random number generation capability is described in detail in section **3.2.11 Cryptographic Random Number Generation** Each TRNG request generates a 128-bit random number. Keys of lengths greater than 128-bits are created by multiple random number generations.

The Secure Cryptographic Engine is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.10 Cryptographic KeyStore

The platform provides the application with a way to store ***cryptographic keys*** such that not even the application can compromise the ***authenticity, integrity, or confidentiality*** of this data. This data can be used for the cryptographic operations ***listed in section 3.2.8 Cryptographic Operation***.

Conformance rationale:

The platform implements a secure key store by leveraging the secure key handling capabilities of the Secure Crypto Engine (SCE9). An MCU block diagram highlighting the SCE9 is shown in ***Figure 3 Secure Crypto Engine***. The "Key Handling" block inside the Secure Crypto Engine ensures that the cryptographic operations taking place inside the SCE9 do not require any sensitive information to be exposed on a CPU- or externally-accessible bus.

The cryptographic operations of the platform are designed to work with wrapped keys. The Renesas proprietary key wrapping algorithms consist of encryption combined with authentication, which ensures confidentiality, integrity, and authenticity of the wrapped key. Keys that have been generated by or installed using the SCE9 are wrapped using an algorithm that incorporates the MCU's Hardware Unique Key (HUK). This ensures that even if the wrapped key is copied to another MCU, it cannot be successfully used by code running on that MCU, since only the Secure Crypto Engine of the MCU that wrapped the key can internally unwrap and use that key.

Renesas do not provide a mechanism to unwrap a wrapped key and provide the plaintext key outside of the Secure Crypto Engine; therefore, an application running on the same MCU that wrapped the key is not able to compromise the authenticity, integrity, nor confidentiality of the key. Keys can be further protected by utilising TrustZone isolation to isolate the key store from the application and by locking the flash blocks that contain any root keys to make them immutable.

Note that for compatibility with legacy systems, the platform supports the capability to use plaintext keys as well as wrapped keys. It is the responsibility of the application developer to properly utilise the secure key installation, generation, and storage capabilities of the platform to fully implement this SFR. Renesas provide software stacks and libraries that support wrapped key usage for various solutions, including a PSA Crypto API implementation and TLS stack. That software is out of scope for this evaluation.

The Secure Cryptographic Engine is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

3.2.11 Cryptographic Random Number Generation

The platform provides the application with a way based on ***thermal noise*** to generate random numbers to as specified in ***SP800-90A and SP800-90B***.

Conformance rationale:

The platform implements random number generation by utilising the True Random Number generation capability of the Secure Crypto Engine inside the MCU. The TRNG implementation consists of an SP800-90A compliant DRBG that is fed by an SP800-90B compliant seed, which is generated from an SP800-22A compliant entropy source. Each TRNG request generates a 128-bit random number.

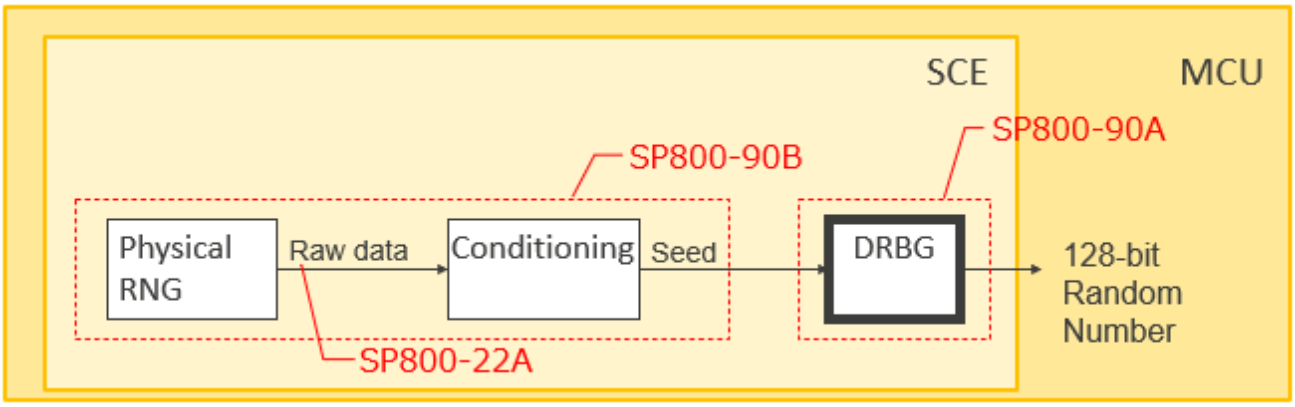

Figure 5 True Random Number Generation

The silicon implementation of the IPs that perform all aspects of true random number generation in the SCE9 Secure Crypto Engine is identical to that in Renesas's earlier SCE7 Secure Crypto Engine, which is included in Renesas RA6M1/2/3 and Synergy S5 and S7 Series MCUs. The silicon implementation of the IP that generates the SP800-22A compliant entropy source has been tested against and passed the NIST Statistical Test Suite version 2.1.1, and found to have an entropy of 7.999987 bits per byte when tested with FourmiLab's Pseudorandom Number Sequence Test Program. The silicon implementation of the IP that performs the SP800-90A compliant DBRG has passed NIST CAVP certification for the SCE7:

- Validation list: DRBG
- Certification number: 2152
- Certification link: https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=26978

The SCE9 DRBG is planned to be submitted explicitly for certification and is expected to pass.

The Secure Cryptographic Engine is thoroughly tested by simulation during the design phase and by device characterisation of the actual silicon. Each die undergoes production test prior to shipment to ensure proper functionality of each MCU.

4 Mapping and sufficiency rationales

4.1 SESIP1 sufficiency

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 Functional Requirements"	All SFRs in this ST are taken from [SESIP]. "Verification of Platform Identity" is included. "~~Secure Update of Platform~~" is struck and documented as required.
	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 are defined in Security Functional Requirements.
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	• **R01UH0892EJ0110** *RA4M2 Group Hardware User's Manual* • **R20AN0577EG0101**	These documents describe the operation of the platform and how to use the platform to

		Renesas RA Family Arm TrustZone Tooling Primer • **R11AN0467EU0100** *Renesas RA Family Security Design with Arm TrustZone IP Protection Project* • **R01AN5562EJ0100** *Standard Boot Firmware for the RA family MCUs Based on Arm® Cortex®-M33*	create an end application.
	AGD_PRE.1 Preparative procedures	• **R20UT4813EJ0100** *Renesas Flash Programmer v3.08 Flash memory programming software User's Manual* • **R11AN0469EU0100** *Renesas RA Family Device Lifecycle Management Key Installation Project* • **R11AN0468EU0100** *Renesas RA Family Securing Data at Rest using Arm® TrustZone® Project*	These documents describe how to prepare the platform as part of an end product.
ALC: Life-cycle support	ALC_FLR.2 Flaw reporting procedures	Section "Flaw Reporting Procedure (ALC_FLR.2)"	The flaw reporting and remediation procedure is described.
AVA_VAN.1	AVA_VAN.1 Vulnerability survey	Section "Vulnerability Survey (AVA_VAN.1)"	The vulnerability survey and associated test results are described.

5 References

[ARM PSA L1] ARM PSA Certified™ Level I Questionnaire, Document reference JSADEN0001, Version 2.0, dated 30/10/2019

[SESIP] GlobalPlatform Technology, Security Evaluation Standard for IoT Platforms (SESIP), GP_FST_070, Public Release v1.0, March 2020