Renesas Electronics Corporation

RA6M4 Security Management Module

FIPS 140-3 Non-Proprietary Security Policy
Revision 1.02

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the RA6M4 Security Management Module from Renesas Electronics Corporation. It describes how the module meets the requirements of Security Level 2 as defined in FIPS 140-3.

1.2 Security Levels

Section	Title	Security Level
1	General	2
2	Cryptographic module specification	2
3	Cryptographic module interfaces	2
4	Roles, services, and authentication	3
5	Software/Firmware security	2
6	Operational environment	N/A
7	Physical security	2
8	Non-invasive security	N/A
9	Sensitive security parameter management	3
10	Self-tests	3
11	Life-cycle assurance	3
12	Mitigation of other attacks	N/A
	Overall Level	2

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

RA6M4 Security Management Module is a hardware cryptographic module certified at FIPS 140-3 Security Level 2. RA6M4 is general-purpose microcontroller designed for industrial, IOT, and infrastructure applications, embedded with robust security features. This module is provided as a reference for developing FIPS 140-3-compliant modules using the RA6M4. By referring to the design of this module, it becomes easier to address FIPS requirements when developing cryptographic modules using the RA6M4 in customer environments.

Module Type: Hardware

Module Embodiment: Single Chip

Module Characteristics :
N/A for this module.

Cryptographic Boundary:

The cryptographic boundary is defined as the perimeter of the IC package as shown in Figure 1.

Figure 1 RA6M4 (R7FA6M4AF3CFB) Chip

Tested Operational Environment's Physical Perimeter (TOEPP) :

The block diagram of the RA6M4 Security Management Module is shown Figure 2. This module provides the services required by FIPS 140-3 and can execute those services by entering string commands via UART. The services provided by the module are implemented in Crypto Firmware. When developing FIPS 140-3 compliant products using RA6M4, implementing only the User Application part can make the certification process easier.

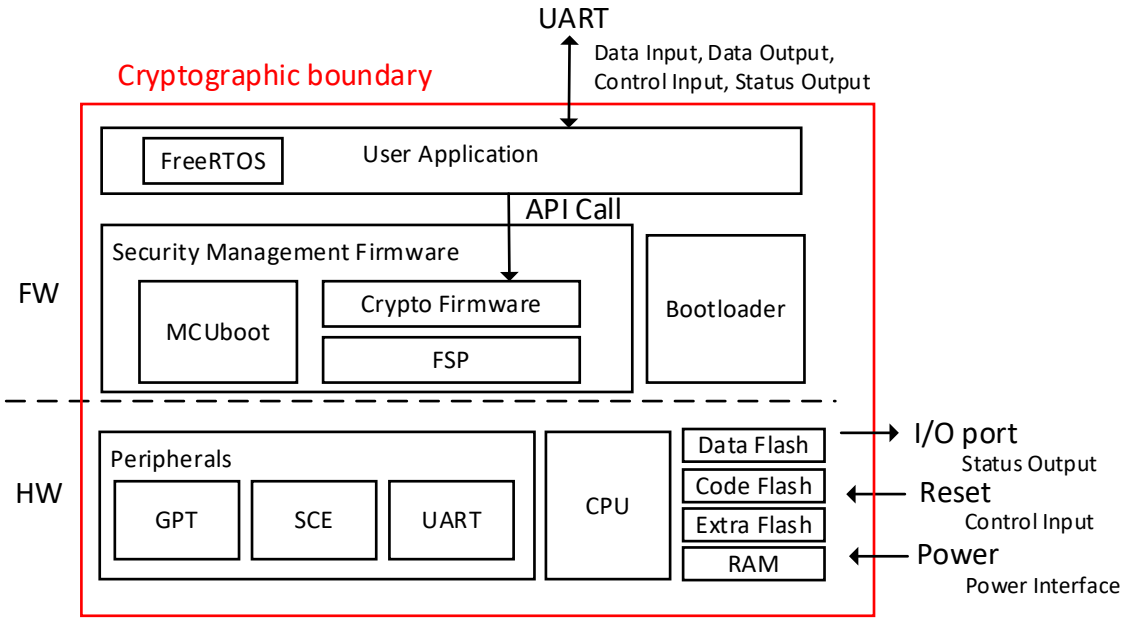

Figure 2 Block Diagram

Component	Description
FW	
User Application	Parses serial data received via UART and performs the corresponding cryptographic service.

Component		Description
		This module is designed to serve as a reference for developing cryptographic modules that conform to CMVP, requiring modifications only to the User Application.
	FreeRTOS	Task scheduling
Security Management Firmware		Implement functionality that complies with FIPS 140-3 requirements, using the Crypto Firmware API as its interface.
	Crypto Firmware	Provides the following services to meet FIPS 140-3 requirements - FSM - User authentication - Module version and status display - Cryptographic services - Self-test
	FSP	RA Flexible Software Package
	MCUboot	For pre-operational self-test, FW Load Test, and part of CASTs
Boot Loader		For module initialization.
HW		
Peripherals		-
	GPT (General PWM Timer)	For periodic self-test
	SCE (Secure Crypto Engine)	For cryptographic services
	UART (Universal Asynchronous Receiver/Transmitter)	For data input/output, control input and status output.
CPU		Arm Cortex-M33
Data Flash		Flash memory for data store
Code Flash		Flash memory for code store
Extra Flash		Access Restricted Flash Memory
RAM		Random Access Memory

*CAVP certified (A5199) SCE9 Protected Mode v1.1.0 consists of SCE, and FSP that is included in Security Management Firmware version 1.01.

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
RA6M4 Security Management Module	R7FA6M4AF3CFB	Security Management Firmware version:1.01; User Application version:1.00; Boot loader version:1.6.25	Arm Cortex-M33	

Table 2: Tested Module Identification – Hardware

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

2.3 Excluded Components

- Boot Loader
  The Boot Loader is used only during the initialization of the module, and after the initialization is complete, it can't be used permanently due to the Device Lifecycle Management capability of the RA6M4. Therefore, it is an excluded firmware component.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Approved security services are available.	Approved	The successful completion of a service

Table 3: Modes List and Description

The module starts as an approved mode of operation after starting up in Single-Chip mode. To start in Single-Chip mode, set the voltage on the MD pin of RA6M4 to "High" and release reset.

Mode Change Instructions and Status :

N/A for this module.

Degraded Mode Description :

N/A for this module.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5199	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CCM	A5199	Key Length - 128, 256	SP 800-38C
AES-CMAC	A5199	Direction - Generation, Verification Key Length - 128, 256	SP 800-38B
AES-ECB	A5199	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-GCM	A5199	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 256	SP 800-38D
Conditioning Component AES-CBC-MAC SP800-90B	A5199	Key Length - 128	SP 800-90B
Counter DRBG	A5199	Prediction Resistance - No Mode - AES-128 Derivation Function Enabled - No	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5199	Curve - P-224, P-256 Secret Generation Mode - extra bits	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5199	Curve - P-224, P-256 Hash Algorithm - SHA2-256 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5199	Curve - P-224, P-256 Hash Algorithm - SHA2-256	FIPS 186-5
HMAC-SHA2-256	A5199	Key Length - Key Length: 256	FIPS 198-1
KDF SP800-108	A5199	KDF Mode - Counter Supported Lengths - Supported Lengths: 128, 256	SP 800-108 Rev. 1
RSA KeyGen (FIPS186-5)	A5199	Key Generation Mode - probable Modulo - 2048 Primality Tests - 2pow100 Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A5199	Modulo - 2048 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A5199	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
SHA2-256	A5199	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric and Asymmetric	N/A	SP800-133r2 Section 4, example1

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
AES-128 CBC-MAC	no security claimed	For interoperability, performed along with inputting User key

Table 6: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES ECB/CBC encryption/decryption	BC-UnAuth	AES ECB/CBC encryption/decryption	key strength:128/256 bit	AES-CBC: (A5199) AES-ECB: (A5199)
AES CCM/GCM encryption/decryption	BC-Auth	AES CCM/GCM encryption/decryption	key strength:128/256 bit	AES-CCM: (A5199) AES-GCM: (A5199)
AES CMAC	MAC	AES CMAC generation / verification	key strength:128/256 bit	AES-CMAC: (A5199)
RSA Sig Gen	DigSig-SigGen	RSA Signature generation		RSA SigGen (FIPS186-5): (A5199) SHA2-256: (A5199)
RSA Sig Ver	DigSig-SigVer	RSA Signature verification	key length:2048/3072/4096 bit	RSA SigVer (FIPS186-5): (A5199) SHA2-256: (A5199)
SHA	SHA	Sha message digest		SHA2-256: (A5199)

Name	Type	Description	Properties	Algorithms
ECDSA Sig Gen	DigSig-SigGen	ECDSA Signature generation	curve:P-224. P-256	ECDSA SigGen (FIPS186-5): (A5199) SHA2-256: (A5199)
ECDSA Sig Ver	DigSig-SigVer	ECDSA Signature verification	curve:P-224. P-256	ECDSA SigVer (FIPS186-5): (A5199) SHA2-256: (A5199)
HMAC	MAC	HMAC generation /verification	key length:256 bit	HMAC-SHA2-256: (A5199)
RSA 2048 Key Pair Generation	AsymKeyPair-KeyGen CKG	RSA 2048 Key Pair Generation	key length:2048 bit	RSA KeyGen (FIPS186-5): (A5199) CKG: ()
ECDSA Key Pair Generation	AsymKeyPair-KeyGen CKG	ECDSA key pair generation	curve:P-224, P-256	ECDSA KeyGen (FIPS186-5): (A5199) CKG: ()
Random Number Generation	DRBG	Random number generation		Counter DRBG: (A5199) Conditioning Component AES-CBC-MAC SP800-90B: (A5199)
FW update	UNK	Update User Application area	curve:P-256	ECDSA SigVer (FIPS186-5): (A5199)
User Authentication	UNK	Authenticate users with a challenge response	curve:P-256	ECDSA SigVer (FIPS186-5): (A5199) Counter DRBG: (A5199)

Name	Type	Description	Properties	Algorithms
AES 128 Key Generation	CKG	AES 128 Key Generation	key strength:128 bit	CKG: ()
KEK for HUK deriving	KBKDF	Derive a KEK for HUK	key strength:256 bit	KDF SP800-108: (A5199)
User key decryption for key update	BC-UnAuthDecrypt	Decryption of user keys for input to the module	key strength:128 bit	AES-CBC: (A5199) AES-128 CBC-MAC: ()
KTS User key wrapping	KTS-Wrap	Encryption of user keys for secure storage in the module	Standard:N/A IG D.G:approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-CCM: (A5199)
KTS User key unwrapping	KTS-Unwrap	Decryption of user keys for output from the module	Standard:N/A IG D.G:approved method from IG D.G Caveat:Key establishment methodology provides 256 bits of security strength	AES-CCM: (A5199)

Table 7: Security Function Implementations

Although the module supports the AES 256-bit operations, it imports the key for the operation externally and cannot generate the key internally.
Similarly, although the module supports RSA operations for 3072 and 4096-bit sizes, it imports the keys for these operations externally and cannot generate the keys internally.

2.7 Algorithm Specific Information

- AES-GCM encryption is in compliance with IG.C.H Scenario 2. When the AES-GCM encryption service is executed, a 96-bit IV is generated from the approved DRBG inside the cryptographic module. This specification assures that the probability of being called with the same key and the same IV is less than 2^32.
- RSA Signature Generation/Verification is in compliance with IG C.F. RSA signature generation supports a modules length of 2048 bits, and RSA signature verification supports modules length of 2048, 3072 and 4096 bits.

2.8 RBG and Entropy

Cert Number	Vendor Name
E176	Renesas Electronics Corporation

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Entropy Source	Physical	R7FA6M4AF3CFB	128	128	A5199

Table 9: Entropy Sources

Entropy Information:
The module has an analog noise source and conditions it using AES-128 CBC-MAC for output. Each 128-bit output sample provides a minimum entropy of 128 bits, which means full entropy.

RNG Information:
Use the entropy source output as a seed to generate random numbers with AES-128 CTR-DRBG.

2.9 Key Generation

This module uses a DRBG compliant with SP800-90A Rev.1 to generate AES, RSA and ECDSA keys. The random numbers used for the cryptographic keys are obtained from the DRBG. It performs cryptographic key generation(CKG) in accordance with SP800-133r2 Section 4, example1. CKG uses the output values of the CTR-DRBG described in NIST SP800-90A as keys.

2.10 Key Establishment

N/A for this module.

2.11 Industry Protocols

N/A for this module.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
UART TXD	Data Output Status Output	Output of Service Execution
UART RXD	Data Input Control Input	Input for Service Execution
P415	Status Output	Set High means that Operator is authenticated.
P404	Status Output	Set High means that zeroization is completed.
P400	Status Output	This port is error indicator, TogglePO Self-Test Error, Set HighCO Self-Test ERROR
Reset	Control Input	Reset
Clock	Control Input	Clock
Power	Power	3.3V power

Table 10: Ports and Interfaces

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Challenge response	The challenge-response method is implemented for user authentication. When the user authentication service is executed, the module sends a 128-bit challenge. The ECDSA P-256 signature is sent back in response to this challenge, and if the signature is correct, the authentication is successful.	User Authentication	The target strength I 1/1,000,000. The probability that a random attempt will succeed, or a false acceptance will occur depends on the ECDSA P-256 strength. Therefore, the probability is 1/2^128, which is less than 1/1,000,000.	The target strength I 1/100,000. Since the module generates a 1 ms wait for each authentication attempt. Therefore, the probability that multiple attacks within a given minute will be successful is 60,000/2^128, which is less than 1/100,000.

Table 11: Authentication Methods

This module provides identity-based authentication, with the authentication method embedded in challenge response authentication using 128-bit random number. It stores authentication information, including ID, role and authentication public key, linked together. Only one user can be logged in at the same time.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
CO	Identity	Crypto Officer	Challenge response
User	Identity	User	Challenge response

Table 12: Roles

This module supports two roles: User and CO(Crypto Officer) role. The maintenance role isn't implemented.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
AES encryption	Perform AES ECB/CBC/ CCM/GCM encryption	"aes enc result:", "aesccm enc result:", "aesgcm enc result:" via UART	Wrapped AES-KEY; plain text	cipher text	AES ECB/CBC encryption/ decryption AES CCM/GCM encryption/ decryption KTS User key unwrapping	CO - AES-KEY: W,E - HUK: E User - AES-KEY: W,E - HUK: E
AES decryption	Perform AES ECB/CBC/ CCM/GCM decryption	"aes dec result:", "aesccm dec result:", "aesgcm dec result:" via UART	Wrapped AES-KEY; cipher text	plain text	AES ECB/CBC encryption/ decryption AES CCM/GCM encryption/ decryption KTS User key unwrapping	CO - AES-KEY: W,E - HUK: E User - AES-KEY: W,E - HUK: E
AES CMAC Generation	Perform AES CMAC generation	"MAC:" via UART	Wrapped AES-KEY; message	MAC value	AES CMAC KTS User key unwrapping	CO - AES-KEY: W,E - HUK: E User - AES-KEY: W,E - HUK: E
AES CMAC Verification	Perform AES CMAC verification	"MAC:" via UART	Wrapped AES-KEY; message; MAC value	"True" or "False"	AES CMAC KTS User key unwrapping	CO - AES-KEY: E - HUK: E User - AES-KEY: E - HUK: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
RSA Signature Generation	Perform RSA Signature Generation	"signature:" via UART	Wrapped RSA-PRIVATE-KEY; message	signature	RSA Sig Gen KTS User key unwrapping	CO - RSA-PRIVATE-KEY: W,E - HUK: E User - RSA-PRIVATE-KEY: W,E - HUK: E
RSA Signature Verification	Perform RSA Signature Verification	"signature:" via UART	Wrapped RSA-PUBLIC-KEY; message; signature	"True" or "False"	RSA Sig Ver KTS User key unwrapping	CO - RSA-PUBLIC-KEY: W,E - HUK: E User - RSA-PUBLIC-KEY: W,E - HUK: E
ECDSA Signature Generation	Perform ECDSA Signature Generation	"signature:" via UART	Wrapped ECDSA-PRIVATE-KEY; message	signature	ECDSA Sig Gen KTS User key unwrapping	CO - ECDSA-PRIVATE-KEY: W,E - HUK: E User - ECDSA-PRIVATE-KEY: W,E - HUK: E
ECDSA Signature Verification	Perform ECDSA Signature Verification	"signature:" via UART	Wrapped ECDSA-PUBLIC-KEY; message; signature	"True" or "False"	ECDSA Sig Ver KTS User key unwrapping	CO - ECDSA-PUBLIC-KEY: E - HUK: E User - ECDSA-PUBLIC-KEY: E - HUK: E
Random number generation	Perform Random number generation	"random number:" via UART	-	Random numbers	Random Number Generation	CO - DRBG internal state (V, Key): W,E User - DRBG internal state (V, Key): W,E
HMAC Generation	Perform HMAC Generation	"MAC:" via UART	Wrapped HMA	MAC value	HMAC KTS User	CO - HMAC-256-KEY: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			C-256-KEY; message		key unwrapping	- HUK: E User - HMAC-256-KEY: W,E - HUK: E
HMAC Verification	Perform HMAC Verification	"MAC:" via UART	Wrapped HMAC-256-KEY; message; MAC value	"True" or "False"	HMAC KTS User key unwrapping	CO - HMAC-256-KEY: W,E - HUK: E User - HMAC-256-KEY: W,E - HUK: E
SHA Hash Generation	Perform SHA Hash Generation	"message digest:" via UART	message	digest	SHA	CO User
AES 128 key generation	Perform AES 128 key generation	"Wrapped AES-128 key:" via UART	-	Wrapped AES-KEY	AES 128 Key Generation KTS User key wrapping	CO - DRBG internal state (V, Key): W,E - AES-KEY: G - HUK: E
RSA 2048 key pair generation	Perform RSA 2048 key pair generation	"Wrapped rsa-2048 private key:" and "Wrapped rsa-2048 public key:" via UART	-	Wrapped RSA-PRIVATE-KEY; Wrapped RSA-PUBLIC-KEY	RSA 2048 Key Pair Generation KTS User key wrapping	CO - DRBG internal state (V, Key): W,E - RSA-PRIVATE-KEY: G - RSA-PUBLIC-KEY: G - HUK: E
ECDSA P224 key pair generation	Perform ECDSA P224 key pair generation	"Wrapped ecdsa-p224 private key:" and "Wrapped ecdsa-p224 public key:" via UART	-	Wrapped ECDSA-PRIVATE-KEY; Wrapped ECDSA-	ECDSA Key Pair Generation KTS User key wrapping	CO - DRBG internal state (V, Key): W,E - ECDSA-PRIVATE-KEY: G - ECDSA-PUBLIC-KEY: G - HUK: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
				PUBLIC-KEY		
ECDSA P256 key pair generation	Perform ECDSA P256 key pair generation	"Wrapped ecdsa-p256 private key:" and "Wrapped ecdsa-p256 public key:" via UART	-	Wrapped ECDSA-PRIVATE-KEY; Wrapped ECDSA-PUBLIC-KEY	ECDSA Key Pair Generation KTS User key wrapping	CO - DRBG internal state (V, Key): W,E - ECDSA-PRIVATE-KEY: G - ECDSA-PUBLIC-KEY: G - HUK: E
AES 128 key update	Perform AES 128 key update	"Wrapped AES_128_KEY:" via UART	encrypted AES-KEY	Wrapped AES-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - AES-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
AES 256 key update	Perform AES 256 key update	"Wrapped AES_256_KEY:" via UART	encrypted AES-KEY	Wrapped AES-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - AES-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
RSA 2048 private key update	Perform RSA 2048 private key update	"Wrapped RSA_PRIVATE_KEY:" via UART	encrypted RSA-PRIVATE-KEY	Wrapped RSA-PRIVATE-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - RSA-PRIVATE-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
RSA 2048 public key update	Perform RSA 2048 public key update	"Wrapped RSA_PUBLIC_KEY:" via UART	encrypted RSA-PUBLIC-KEY	Wrapped RSA-PUBLIC-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - RSA-PUBLIC-KEY: W - HUK: E - AES-CBC-MAC-KEY: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
RSA 3072 public key update	Perform RSA 3072 public key update	"Wrapped RSA_3072_PUBLIC_KEY:" via UART	encrypted RSA-PUBLIC-KEY	Wrapped RSA-PUBLIC-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - RSA-PUBLIC-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
RSA 4096 public key update	Perform RSA 4096 public key update	"Wrapped RSA_4096_PUBLIC_KEY:" via UART	encrypted RSA-PUBLIC-KEY	Wrapped RSA-PUBLIC-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - RSA-PUBLIC-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
ECDSA P224 private key update	Perform ECDSA P224 private key update	"Wrapped ECDSA_P224_PRIVATE_KEY:" via UART	encrypted ECDSA-PRIVATE-KEY	Wrapped ECDSA-PRIVATE-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - ECDSA-PRIVATE-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
ECDSA P224 public key update	Perform ECDSA P224 public key update	"Wrapped ECDSA_P224_PUBLIC_KEY:" via UART	encrypted ECDSA-PUBLIC-KEY	Wrapped ECDSA-PUBLIC-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - ECDSA-PUBLIC-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
ECDSA P256 private key update	Perform ECDSA P256 private key update	UART sends strings that "Wrapped ECDSA_P256_PRIVATE_KEY:"	encrypted ECDSA-PRIVATE-KEY	Wrapped ECDSA-PRIVATE-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - ECDSA-PRIVATE-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
ECDSA P256 public key update	Perform ECDSA P256 public key update	"Wrapped ECDSA_P256_PUBLIC_KEY:" via UART	encrypted ECDSA-PUBLIC-KEY	Wrapped ECDSA-PUBLIC-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - ECDSA-PUBLIC-KEY: W - HUK: E - AES-CBC-MAC-KEY: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
HMAC 256 key update	Perform HMAC 256 key update	"Wrapped HMAC_256_KEY :" via UART	encrypted HMAC-256-KEY	Wrapped HMAC-256-KEY	User key decryption for key update KTS User key wrapping	CO - KEY-UPDATE-KEY: E - HMAC-256-KEY: W - HUK: E - AES-CBC-MAC-KEY: E
Show module version	Show current module status, logged-in user and error log for self-test.	none	-	module id / version	None	CO User Unauthenticated
Show module status	Show module status	none	-	module status	None	CO User Unauthenticated
Zeroization	Perform CSP Zeroization	P404 Set High	-	-	None	CO - RTL Key: Z - MCUBOOT_IMAGE_SIGNING_KEY: Z
Firmware update	Perform Firmware update	none	new firmware; signature	-	FW update	CO - MCUBOOT_IMAGE_SIGNING_KEY: E
User authentication	Challenge response authentication using ECDSA P-256 Signature verification	P415 Set High	User ID; signature	-	User Authentication	Unauthenticated - ECDSA-P256-USER-PUBLIC-KEY: E - ECDSA-P256-CO-PUBLIC-KEY: E - DRBG internal state (V, Key): W,E - HUK: E
Add User	Perform Add User	none	User ID; encrypted ECDSA-P256-USE	-	User key decryption for key update	CO - ECDSA-P256-USER-PUBLIC-KEY: W - KEY-UPDATE-KEY: E - HUK: E - AES-CBC-MAC-KEY: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			R-PUBLIC-KEY			
Delete User	Perform Delete User	none	User ID	-	None	CO - ECDSA-P256-CO-PUBLIC-KEY: Z
pre operational self test	Perform pre operational self test	none	-	-	KEK for HUK deriving	CO - RTL Key: E - KEK for HUK: G,E - HUK: W - MCUBOOT_IMAGE_SIGNING_KEY: E User - RTL Key: E - KEK for HUK: G,E - HUK: W - MCUBOOT_IMAGE_SIGNING_KEY: E Unauthenticated - RTL Key: E - KEK for HUK: G,E - HUK: W - MCUBOOT_IMAGE_SIGNING_KEY: E
On-demand selftest	Perform ondemand conditional self test	"Conditional test completed" via UART	-	-	KEK for HUK deriving	CO - RTL Key: E - KEK for HUK: G,E,Z - HUK: W - DRBG seed: G - DRBG internal state (V, Key): G,W,E User - RTL Key: E - KEK for HUK: G,E,Z - HUK: W - DRBG seed: G - DRBG internal state (V, Key): G,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Open	Initialize the module	"Crypto Firmware initialized successfully" via UART	-	-	KEK for HUK deriving	CO - RTL Key: E - KEK for HUK: G,E,Z - HUK: W - DRBG seed: G - DRBG internal state (V, Key): G,W,E User - RTL Key: E - KEK for HUK: G,E,Z - HUK: W - DRBG seed: G - DRBG internal state (V, Key): G,W,E Unauthenticated - RTL Key: E - KEK for HUK: G,E,Z - HUK: W - DRBG seed: G - DRBG internal state (V, Key): G,W,E
Close	Close the module	none	-	-	None	CO User Unauthenticated
Display User List	Display User List	none	-	-	None	CO User

Table 13: Approved Services

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

The Firmware load function via UART interface is implemented. ECDSA P-256 signature verification is performed during the firmware load test. The update applies only to the User

Application, and other firmware cannot be updated. To maintain certification, the firmware to be loaded must be validated.

5 Software/Firmware Security

5.1 Integrity Techniques

The firmware integrity test is executed after the module's power off/on (this may also mean reset hereafter). It is performed by MCUboot according to the following procedures.

- MCUboot's integrity is verified by EDC(CRC32).
- The User Application and Crypto Firmware area are verified by ECDSA P-256.

5.2 Initiate on Demand

This module can execute an on-demand test by executing the power off/on or selftest command.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied :

N/A for this module.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Tamper-evident coating on chip	Every month	Confirmation that there is no visual damage

Table 14: Mechanisms and Actions Required

The chip is treated with a opaque coating designed to provide tamper evidence.
The module consists of standard, production-quality ICs, designed to meet commercial-grade specifications, and uses standard passivation techniques for the entire chip.

Figure 3 Tamper-evident Coating on Chip

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Code Flash	Flash memory for code store	Static
Data Flash	Flash memory for data store	Static
Extra Flash	Access Restricted Flash Memory	Static
SCE Internal register	Internal registers of SCE	Dynamic
RAM	RAM	Dynamic

Table 15: Storage Areas

The Bootloader, which is an excluded component, is stored in Extra Flash.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Key update (input)	UART	Data Flash	Encrypted	Manual	Electronic	User key decryption for key update
Key update (output)	Data Flash	UART	Encrypted	Automated	Electronic	KTS User key wrapping
Cryptographic service parameter (input)	UART	RAM	Encrypted	Automated	Electronic	KTS User key unwrapping

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Add User (input)	UART	Data Flash	Encrypted	Manual	Electronic	User key decryption for key update
Key generation (output)	SCE Internal register	UART	Encrypted	Automated	Electronic	KTS User key wrapping
Initialization Process (input)	UART	Data Flash	Encrypted	Manual	Electronic	User key decryption for key update

Table 16: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroization Service	Erase the Code Flash area containing the SCE driver that includes the 'RTL Key', which serves as the root of SSP protection.	Overwrite the flash area with 0xFF.	CO
power off/on	The SSP held in the SCE internal register is volatilized by cutting power with power off/on. Additionally, when the initialization process is performed, SCE internal register is set to "0". Turning the power off and on triggers the initialization process as zeroization.	Clear SCE internal register to 0	CO, User

Table 17: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RTL Key	AES-128 key used for deriving the KEK for HUK	128 - 128	Symmetric Key - CSP	Generated at factory		KEK for HUK deriving
KEK for HUK	AES-256 key used to encrypt the HUK for storage in the RA6M4	256 - 256	Symmetric Key - CSP	KEK for HUK deriving		AES CCM/GCM encryption/decryption
HUK	Hardware Unique Key for each device	256 - 256	Symmetric Key - CSP	Generated at factory		KTS User key wrapping KTS User key unwrapping

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG internal state (V, Key)	DRBG internal state	256 - 128	DRBG intrnal state - CSP	Counter DRBG (A5199)		Random Number Generation Counter DRBG (A5199)
DRBG seed	Seed used for DRBG Instantiation.	256 - 256	DRBG seed - CSP	Counter DRBG (A5199)		Counter DRBG (A5199)
KEY-UPDATE-KEY	Key used to update the user key	128 - 128	Symmetric Key - CSP	Generated outside the module		User key decryption for key update
AES-KEY	128/256-bit key used for AES encryption/decryption in ECB/CBC/GCM/CCM modes and for generating/verifying AES CMAC	128, 256 - 128, 256	Symmetric Key - CSP	AES 128 Key Generation		AES ECB/CBC encryption/decryption AES CCM/GCM encryption/decryption AES CMAC
RSA-PRIVATE-KEY	RSA private key used for signature generation	2048 - 112	Private Key - CSP	RSA 2048 Key Pair Generation		RSA Sig Gen
RSA-PUBLIC-KEY	RSA public key used for signature verification	2048, 3072, 4096 - 112, 128, 150	Public Key - PSP	RSA 2048 Key Pair Generation		RSA Sig Ver
ECDSA-PRIVATE-KEY	ECDSA private key used for signature generation	224, 256 - 112, 128	Private Key - CSP	ECDSA Key Pair Generation		ECDSA Sig Gen

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDSA-PUBLIC-KEY	ECDSA public key used for signature verification	224, 256 - 112, 128	Public Key - PSP	ECDSA Key Pair Generation		ECDSA Sig Ver
ECDSA-P256-CO-PUBLIC-KEY	ECDSA public key for Crypto Officer used for user authentication	256 - 128	Public Key - PSP	Generated outside the module		User Authentication
ECDSA-P256-USER-PUBLIC-KEY	ECDSA public key for User used for user authentication	256 - 128	Public Key - PSP	Generated outside the module		User Authentication
HMAC-256-KEY	HMAC-SHA2-256 Key used for generation/verification	256 - 256	Symmetric Key - CSP	Generated outside the module		HMAC
MCUBOOT_IMAGE_SIGNING_KEY	ECDSA public key for the firmware integrity test and the load test	256 - 128	Public Key - PSP	Generated outside the module		FW update
AES-CBC-MAC-KEY	Key used for interoperability with inputting User key	128 - -	Symmetric Key - Neither	Generated outside the module		AES-128 CBC-MAC

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
RTL Key		Code Flash:Obfuscated		Zeroization Service	
KEK for HUK		SCE Internal register:Plaintext	Until power off/on	power off/on	RTL Key:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					HUK:Decrypts
HUK		Extra Flash:Encrypted		N/A	AES-KEY:Encrypts RSA-PRIVATE-KEY:Encrypts RSA-PUBLIC-KEY:Encrypts ECDSA-PRIVATE-KEY:Encrypts ECDSA-PUBLIC-KEY:Encrypts ECDSA-P256-CO-PUBLIC-KEY:Encrypts ECDSA-P256-USER-PUBLIC-KEY:Encrypts HMAC-256-KEY:Encrypts KEY-UPDATE-KEY:Decrypts AES-KEY:Decrypts RSA-PRIVATE-KEY:Decrypts RSA-

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					PUBLIC-KEY:Decrypts ECDSA-PRIVATE-KEY:Decrypts ECDSA-PUBLIC-KEY:Decrypts ECDSA-P256-CO-PUBLIC-KEY:Decrypts ECDSA-P256-USER-PUBLIC-KEY:Decrypts HMAC-256-KEY:Decrypts AES-CBC-MAC-KEY:Decrypts
DRBG internal state (V, Key)		SCE Internal register:Plaintext	Until power off/on	power off/on	
DRBG seed		SCE Internal register:Plaintext	Until power off/on	power off/on	
KEY-UPDATE-KEY	Initialization Process (input)	Data Flash:Encrypted		N/A	AES-KEY:Decrypts RSA-PRIVATE-KEY:Decrypts RSA-PUBLIC-KEY:Decrypts

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					ECDSA-PRIVATE-KEY:Decrypts ECDSA-PUBLIC-KEY:Decrypts ECDSA-P256-CO-PUBLIC-KEY:Decrypts ECDSA-P256-USER-PUBLIC-KEY:Decrypts HMAC-256-KEY:Decrypts HUK:Encrypted by
AES-KEY	Key update (input) Key update (output) Cryptographic service parameter (input) Key generation (output)	Data Flash:Encrypted		N/A	HUK:Encrypted by
RSA-PRIVATE-KEY	Key update (input) Key update (output) Cryptographic service	Data Flash:Encrypted		N/A	HUK:Encrypted by

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	parameter (input) Key generation (output)				
RSA-PUBLIC-KEY	Key update (input) Key update (output)	Data Flash:Encrypted		N/A	HUK:Encrypted by
ECDSA-PRIVATE-KEY	Key update (input) Key update (output) Cryptographic service parameter (input) Key generation (output)	Data Flash:Encrypted		N/A	HUK:Encrypted by
ECDSA-PUBLIC-KEY	Key update (input) Key update (output) Cryptographic service parameter (input) Key generation (output)	Data Flash:Encrypted		N/A	HUK:Encrypted by
ECDSA-P256-CO-PUBLIC-KEY	Initialization Process (input)	Data Flash:Encrypted		N/A	HUK:Encrypted by
ECDSA-P256-USER-PUBLIC-KEY	Add User (input)	Data Flash:Encrypted		N/A	HUK:Encrypted by

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
HMAC-256-KEY	Key update (input) Key update (output) Cryptographic service parameter (input)	Data Flash:Encrypted		N/A	HUK:Encrypted by
MCUBOOT_IMAGE_SIGNING_KEY	Initialization Process (input)	Data Flash:Plaintext		Zeroization Service	
AES-CBC-MAC-KEY	Initialization Process (input)	Data Flash:Encrypted		N/A	HUK:Encrypted by

Table 19: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
CRC-32	32 bit	Error Detection Code	SW/FW Integrity	Input via UART can be accepted	MCUboot integrity check
ECDSA P256	ECDSA P256	ECDSA P256 signature verification	SW/FW Integrity	Input via UART can be accepted	User Application & Crypto Firmware Integrity check

Table 20: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF SP800-108 (A5199)	256bit	KAT	CAST	"Conditional test completed." via UART	KDF	Execution of Open or Execution of On-demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						selftest or Periodic self-test
AES CCM enc(A5199)	128	KAT	CAST	"Conditional test completed." via UART	Encrypt	Execution of Open or Execution of On-demand selftest or Periodic self-test
AES CCM dec(A5199)	128	KAT	CAST	"Conditional test completed." via UART	Decrypt	Execution of Open or Execution of On-demand selftest or Periodic self-test
Counter DRBG (A5199)	256	KAT	CAST	"Conditional test completed." via UART	instantiate function, generate function	Execution of Open or Execution of On-demand selftest or Periodic self-test
Entropy Source Health Test	-	fault detection test	CAST	"Conditional test completed." via UART	APT, RCT	Execution of Open or Execution of On-demand selftest or Periodic self-test
AES-GCM (A5199)	128	KAT	CAST	"Conditional test completed." via UART	Encrypt	Execution of Open or Execution of On-demand selftest or Periodic self-test
AES-ECB (A5199)	128	KAT	CAST	"Conditional test completed." via UART	Decrypt	Execution of Open or Execution of On-demand selftest or Periodic self-test
RSA SigGen	2048	KAT	CAST	"Conditional test	Sign	Execution of Open or Execution of

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-5) (A5199)				completed." via UART		On-demand selftest or Periodic self-test
RSA SigVer (FIPS186-5) (A5199)	2048	KAT	CAST	"Conditional test completed." via UART	Verify	Execution of Open or Execution of On-demand selftest or Periodic self-test
SHA2-256 (A5199)	256	KAT	CAST	"Conditional test completed." via UART	Hash	Execution of Open or Execution of On-demand selftest or Periodic self-test
HMAC-SHA2-256 (A5199)	256	KAT	CAST	"Conditional test completed." via UART	HMAC Gen, HMAC Verify	Execution of Open or Execution of On-demand selftest or Periodic self-test
ECDSA SigGen (FIPS186-5) (A5199)	P224	KAT	CAST	"Conditional test completed." via UART	Sign	Execution of Open or Execution of On-demand selftest or Periodic self-test
ECDSA SigVer (FIPS186-5) (A5199)	P224	KAT	CAST	"Conditional test completed." via UART	Verify	Before Pre-Operational Self-Tests, and Execution of Open or Execution of On-demand selftest or Periodic self-test
RSA KeyGen (FIPS186-5) (A5199)	2048	PCT	PCT	Output tested key pair via UART.	Key Pair Gen	Execution of RSA 2048 key pair generation
ECDSA KeyGen	224, 256	PCT	PCT	Output tested key	Key Pair Gen	Execution of ECDSA key

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-5) (A5199)				pair via UART.		pair generation services
FW Load Test	ECDSA p256 signature	Signature Verify	SW/FW Load	"fwupdate success" via UART	FW Load Test	Execution of fwupdate command

Table 21: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
CRC-32	Error Detection Code	SW/FW Integrity	per 15 minutes	programmatically
ECDSA P256	ECDSA P256 signature verification	SW/FW Integrity	per 15 minutes	programmatically

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDF SP800-108 (A5199)	KAT	CAST	per 10 minutes	programmatically
AES CCM enc(A5199)	KAT	CAST	per 10 minutes	programmatically
AES CCM dec(A5199)	KAT	CAST	per 10 minutes	programmatically
Counter DRBG (A5199)	KAT	CAST	per 10 minutes	programmatically
Entropy Source Health Test	fault detection test	CAST	per 10 minutes	programmatically
AES-GCM (A5199)	KAT	CAST	per 10 minutes	programmatically
AES-ECB (A5199)	KAT	CAST	per 10 minutes	programmatically
RSA SigGen (FIPS186-5) (A5199)	KAT	CAST	per 10 minutes	programmatically
RSA SigVer (FIPS186-5) (A5199)	KAT	CAST	per 10 minutes	programmatically
SHA2-256 (A5199)	KAT	CAST	per 10 minutes	programmatically
HMAC-SHA2-256 (A5199)	KAT	CAST	per 10 minutes	programmatically

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigGen (FIPS186-5) (A5199)	KAT	CAST	per 10 minutes	programmatically
ECDSA SigVer (FIPS186-5) (A5199)	KAT	CAST	per 10 minutes	programmatically
RSA KeyGen (FIPS186-5) (A5199)	PCT	PCT	-	-
ECDSA KeyGen (FIPS186-5) (A5199)	PCT	PCT	-	-
FW Load Test	Signature Verify	SW/FW Load	-	-

Table 23: Conditional Periodic Information

The Pre-operational self-test measures its interval using a timer and runs automatically every 15 minutes.
The Conditional self-test measures its interval using a timer and runs automatically every 10 minutes.
The timer starts counting once user authentication is successful and the state transitions to either the User or CO role.

The on-demand pre-operational test can be executed by power off/on.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
PO Self-Test Error State	Error state caused by failure of self-test executed by MCUboot	In the case where the self-test executed by MCUboot encounters an error.	power off/on	P400 toggles between low and high at 0.5 second intervals
CO Self-Test Error State	Error state caused by failure of self-test executed by User Application	In the case where the self-test executed by User Application encounters an error.	power off/on	P400 Set High

Table 24: Error States

This module transitions to an error state if it fails the self-test.
MCUboot performs the Po-operational Test, FW Load Test, and ECDSA SigVer (FIPS186-5).
Other self-tests are executed by the User Application.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Renesas Electronics provides the evaluation board for the module, the EK-RA6M4, and the firmware executable file to customers.

- EK-RA6M4
  https://www.renesas.com/en/products/microcontrollers-microprocessors/ra-cortex-m-mcus/ek-ra6m4-evaluation-kit-ra6m4-mcu-group
- Security Management Firmware (ra6m4_security_management_module.zip)

After obtaining the above, please initialize the module following the steps below by CO. The details of the procedure are described in the user setup manual "RA6M4_SMM_Setting_Manual_rev.1.0.docx".
Note that this operation is proceeded with using a NON-networked GPC. Connect the GPC to the module via a serial port of the EK-RA6M4.

1. Set up the EK-RA6M4
2. Writing the executable firmware
3. Writing the user keys
4. Changing the Device Life Cycle
5. Authentication using the default CO public key
6. Updating the default CO public key

The Boot Loader included in the module is written to the Extra Flash area of the RA6M4 at the time of product shipment.
The Boot Loader version can be checked using RFP (Renesas Flash Programmer). By executing a command with the "-sig" option in RFP.

```
rfp-cli -d RA -t jlink -sig
```

The other component version can be outputted by "Show module version" service.

11.2 Administrator Guidance

Refer to the RA6M4_SMM_Usage_Guidance_rev.1.0.docx
Note that using the "Add User" service or using the "key update" services must be proceeded with using a NON-networked GPC connect to the module via a serial port of the EK-RA6M4.

11.3 Non-Administrator Guidance

Refer to the RA6M4_SMM_Usage_Guidance_rev.1.0.docx

11.4 Design and Rules

N/A for this module.

11.5 Maintenance Requirements

N/A for this module.

11.6 End of Life

The following procedures are used to ensure secure sanitization and destruction of cryptographic module.

1. Secure sanitization
   Execute the “zeroization” command in the CO role to zeroize the unprotected SSPs.
2. Secure destruction
   Destroy the RA6M4 (R7FA6M4AF3CFB) to fragments smaller than 2 mm².

12 Mitigation of Other Attacks

N/A for this module.