Samsung Electronics Co., Ltd.

Samsung SCrypto Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

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 Samsung SCrypto Cryptographic Module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

1.2 Security Levels

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

Table 1: Security Levels

1.3 Additional Information

There are three major reasons that a security policy is needed:

- It is required for FIPS 140-3 validation.
- To provide a specification of the cryptographic security that will allow individuals and organizations to determine whether a cryptographic module, as implemented, satisfies a stated security policy.
- To describe to individuals and organizations the capabilities, protection, and access rights provided by the cryptographic module, thereby allowing an assessment of whether the module will adequately serve the individual or organizational security requirements.

This document is part of the package of documents that are submitted for FIPS 140-3 conformance validation of the module. It is intended for the following people:

- Developers.
- FIPS 140-3 testing lab.
- The Cryptographic Module Validation Program (CMVP).
- Administrators of the cryptographic module.
- Users of the cryptographic module.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

Samsung SCrypto Cryptographic Module (hereinafter referred to as “the module”) is a software module implementing general-purpose cryptographic algorithms. The module is running on a multi-chip standalone general-purpose computing platform. The version of the module is 2.7.

The module provides cryptographic services to applications through an application program interface (API). The module also interacts with the operating system via system calls.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Module Characteristics:

Cryptographic Boundary:

The module is defined as a multi-chip standalone software module, with the boundary of the Tested Operational Environment’s Physical Perimeter (TOEPP) being defined as the physical perimeter of the tested platform enclosure around which everything runs.

Tested Operational Environment’s Physical Perimeter (TOEPP):

The physical perimeter is the hardware platform on which the module is installed. The cryptographic boundary of the module is the SCrypto cryptographic module, a single object module file named fipscanister.o, which is linked to create the executable files scrypto_v2.7_x64_qsee_release.a for the tested platform running QSEE 5.24, QSEE 6.1 (64-bit), and scrypto_v2.7_x64_teegris500_sys_release.so for the tested platform running TEEgris 5.0.0 (64-bit).

The Block Diagram figure below illustrates the module’s physical perimeter. The module’s cryptographic boundary consists of all functionalities contained within the module’s compiled source code.

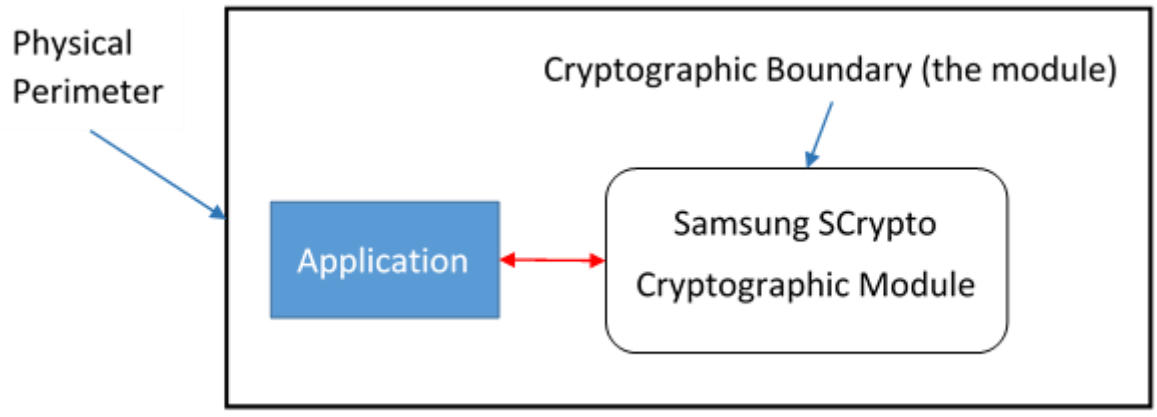

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fipscanister.o	2.7		HMAC-SHA2-256 (A3243)

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
QSEE 5.24 (64-bit)	Samsung Galaxy S23+	Qualcomm Snapdragon 8 Gen 2	No		2.7
QSEE 6.1 (64-bit)	Samsung Galaxy S24	Qualcomm Snapdragon 8 Gen 3	No		2.7
TEEgris 5.0.0 (64-bit)	Samsung Galaxy S24	Samsung Electronics Exynos 2400	No		2.7
TEEgris 5.0.0 (64-bit)	Samsung Galaxy Tab Active 5	Samsung Electronics Exynos 1380	No		2.7
TEEgris 5.0.0 (64-bit)	Samsung Galaxy Tab S9 FE	Samsung Electronics Exynos 1380	No		2.7

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

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

Operating System	Hardware Platform
Linux Kernel 5.15	Samsung Electronics Exynos 1380 running on Samsung A35

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

The module does not claim any excluded components.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	Automatically entered whenever an approved service is requested.	Approved	Return code “1” denotes approved mode of operation
Non-Approved Mode	Automatically entered whenever a non-approved service is requested.	Non-Approved	Return code “0” denotes approved mode of operation

Table 5: Modes List and Description

The module supports both Approved and Non-Approved modes of operation. The module will be in approved mode when all pre-operational self-tests have completed successfully and only approved algorithms/services are invoked. The non-approved mode is entered when a non-approved algorithm/non-approved service is invoked. When the module is initialized, the self-tests are executed automatically. After successful completion of self-tests, the module enters operational state. The module supports only normal operation. Degraded operation is not supported.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3243	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3243	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-524288 Increment 8	SP 800-38B
AES-CTR	A3243	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128 Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A3243	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3243	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 504, 512, 1016, 1024 AAD Length - AAD Length: 504, 512, 1016, 1024	SP 800-38D
AES-KW	A3243	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128, 256, 320	SP 800-38F
AES-OFB	A3243	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
Counter DRBG	A3243	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - No Additional Input - Additional Input: 0, 128, 256, 384 Entropy Input - Entropy Input: 384 Nonce - Nonce: 0	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Personalization String Length - Personalization String Length: 0, 128, 256, 384 Returned Bits - 512	
ECDSA KeyGen (FIPS186-4)	A3243	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3243	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3243	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3243	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
HMAC-SHA-1	A3243	MAC - MAC: 160 Key Length - Key Length: 112, 504, 512, 520, 2048	FIPS 198-1
HMAC-SHA2-224	A3243	MAC - MAC: 224 Key Length - Key Length: 112, 504, 512, 520, 2048	FIPS 198-1
HMAC-SHA2-256	A3243	MAC - MAC: 256 Key Length - Key Length: 112, 504, 512, 520, 2048	FIPS 198-1
HMAC-SHA2-384	A3243	MAC - MAC: 384 Key Length - Key Length: 112, 504, 1024, 1032, 2048	FIPS 198-1
HMAC-SHA2-512	A3243	MAC - MAC: 512 Key Length - Key Length: 112, 504, 1024, 1032, 2048	FIPS 198-1
KDF SP800-108	A3243	KDF Mode - Counter MAC Mode - HMAC-SHA2-512 Supported Lengths - Supported Lengths: 512-4096 Increment 1 Fixed Data Order - After Fixed Data, Before Fixed Data, In the Middle of Fixed Data Counter Length - 16, 24, 32, 8 Supports Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
RSA Decryption Primitive (CVL)	A3243	Modulus Length - 2048	FIPS 186-4
RSA KeyGen (FIPS186-4)	A3243	Key Generation Mode - B.3.3 Modulo - 2048, 3072 Primality Tests - Table C.2 Info Generated By Server - No Public Exponent Mode - Fixed Fixed Public Exponent - 010001 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3243	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072 Hash Pair - Hash Algorithm - SHA2-224	FIPS 186-4
RSA SigVer (FIPS186-4)	A3243	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072 Hash Pair - Hash Algorithm - SHA-1 Public Exponent Mode - Fixed Fixed Public Exponent - 010001	FIPS 186-4
SHA-1	A3243	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A3243	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3243	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A3243	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512	A3243	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Asymmetric	N/A	The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per sections 4 and 5 in SP800-133rev2 (vendor affirmed) and FIPS 140-3 IG D.H. A seed (i.e., the random value) used in asymmetric key generation is a direct output from SP800-90Arev1 CTR_DRBG (A3243)

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
DSA Key Generation [FIPS186-4]	DSA keypair generation
DSA Signature Generation [FIPS186-4]	DSA signature generation
DSA Signature Verification [FIPS186-4]	DSA Signature Verification
KAS-FFC-SSC [SP800-56A Rev 3]	Diffie-Hellman Key Agreement primitive
KAS-ECC-SSC [SP800-56A Rev 3]	EC Diffie-Hellman Key Agreement primitive

Table 8: Non-Approved, Not Allowed Algorithms

Please note that due to the lack of the associated self-tests to DSA, KAS-ECC-SSC and KAS-FFC-SSC algorithms, those algorithms are listed in Non-Approved, Not Allowed Algorithms Table. Those non-approved, not allowed algorithms shall not be used in the Approved Mode of Operation.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption	BC-UnAuth	Encryption with AES		AES-CBC: (A3243) AES-CTR: (A3243) AES-ECB: (A3243) AES-OFB: (A3243)
Symmetric decryption	BC-UnAuth	Decryption with AES		AES-CBC: (A3243) AES-CTR: (A3243) AES-ECB: (A3243) AES-OFB: (A3243)
Authenticated encryption	BC-Auth	Authenticated encryption with AES		AES-GCM: (A3243)

Name	Type	Description	Properties	Algorithms
Authenticated decryption	BC-Auth	Authenticated decryption with AES		AES-GCM: (A3243)
AES key wrapping	KTS-Wrap	Key wrap with AES		AES-KW: (A3243)
AES key unwrapping	KTS-Wrap	Key unwrap with AES		AES-KW: (A3243)
Message digest generation	SHA	Compute a message digest using Secure Hash Algorithms		SHA-1: (A3243) SHA2-224: (A3243) SHA2-256: (A3243) SHA2-384: (A3243) SHA2-512: (A3243)
HMAC Message Authentication Code	MAC	Compute a Message Authentication Code using HMAC		HMAC-SHA-1: (A3243) HMAC-SHA2-224: (A3243) HMAC-SHA2-256: (A3243) HMAC-SHA2-384: (A3243) HMAC-SHA2-512: (A3243) SHA-1: (A3243) SHA2-224: (A3243) SHA2-256: (A3243) SHA2-384: (A3243) SHA2-512: (A3243)
CMAC Message Authentication Code	MAC	Compute a Message Authentication Code using CMAC		AES-CMAC: (A3243)
RSA key pair generation	AsymKeyPair-KeyGen	Key pair generation using RSA		RSA KeyGen (FIPS186-4): (A3243) Counter DRBG: (A3243) CKG: () Key Type: Asymmetric
RSA signature generation	DigSig-SigGen	Digital signature generation using RSA		RSA SigGen (FIPS186-4): (A3243)
RSA signature verification	DigSig-SigVer	Digital signature verification using RSA		RSA SigVer (FIPS186-4): (A3243)
ECDSA key pair generation/verification	AsymKeyPair-KeyGen AsymKeyPair-KeyVer	Key pair generation using ECDSA		ECDSA KeyGen (FIPS186-4): (A3243) Counter DRBG: (A3243) CKG: () Key Type: Asymmetric ECDSA KeyVer (FIPS186-4): (A3243)
ECDSA signature generation	DigSig-SigGen	Digital signature generation using ECDSA		ECDSA SigGen (FIPS186-4): (A3243)
ECDSA signature verification	DigSig-SigVer	Digital signature verification using ECDSA		ECDSA SigVer (FIPS186-4): (A3243)

Name	Type	Description	Properties	Algorithms
Random number generation with Counter DRBG	DRBG	Random number generation using Counter DRBG		Counter DRBG: (A3243)
Key derivation with KBKDF	KBKDF	Key derivation using KBKDF		KDF SP800-108: (A3243)
RSADP primitive	UNK	RSADP primitive generation		RSA Decryption Primitive: (A3243)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

- The AES-GCM IV generation method from AES Cert. #A3243 is in compliance with IG C.H, scenario #2. The DRBG with Cert. #A3243 is called to generate the IV inside the module, and the IV length is 96 bits. The new AES-GCM key will be generated if the module loses power.

- The module was algorithm tested based on the FIPS 186-4 standard Digital Signatures. According to IG C.K, this module is 186-5 compliant as all 186-4 CAVP tests performed are mathematically identical to the 186-5 CAVP tests. The Module does not support 186-4 DSA or RSA X9.31 for Signature Generation or Signature Verification.

2.8 RBG and Entropy

Cert Number	Vendor Name
E67	Qualcomm Technologies, Inc.
E152	Qualcomm Technologies, Inc.
E221	Samsung Electronics Co., Ltd
E224	Samsung Electronics Co., Ltd

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Entropy Source of the Qualcomm(R) Pseudo Random Number Generator - 1	Physical	Qualcomm Technologies, Inc. N/A Snapdragon(R) 8 Gen 2 Mobile Platform	4	0.420625	
Entropy Source of the Qualcomm(R) Pseudo Random Number Generator-2	Physical	Qualcomm Technologies, Inc. N/A Snapdragon(R) 8 Gen 3 Mobile Platform	4	0.342458	
Samsung TRNG Exynos 2400	Physical	Samsung Electronics Exynos 2400	1	0.5	
Samsung TRNG Exynos 1380	Physical	Samsung Electronics Exynos 1380	1	0.5	

Table 11: Entropy Sources

The module employs an Approved SP 800-90Arev1 CTR_DRBG for creation of random numbers with derivation function enabled. The module uses the physical entropy source (ESV

Certs. #E67, #E152, #E221 or #E224) from the operational environment as the source of random numbers for DRBG seeds, as detailed below.

- Qualcomm Snapdragon 8 Gen 2 on Samsung Galaxy S23+ implements E67;
- Qualcomm Snapdragon 8 Gen 3 on Samsung Galaxy S24 implements E152;
- Samsung Electronics Exynos 2400 on Samsung Galaxy S24 implements E152;
- Samsung Electronics Exynos 1380 on Samsung Galaxy Tab Active 5 or Samsung Galaxy Tab S9 FE implements E224

The Entropy Source produces the random numbers from an entropy pool maintained by the underlying Operating System. The module is a software module that contains an approved DRBG that is seeded exclusively from one known entropy source located inside the module's physical perimeter but outside the module's boundary. The module provides at least 256 bits of entropy to instantiate the DRBG. The module performs the Repetition Count Test (RCT) and Adaptive Proportion Test (APT) as the Health Test to the entropy source that is used to instantiate the module's DRBG.

The DRBG is instantiated with 1536, 4-bit samples (a total of 6,144 bits of noise) from the entropy source, which provides at least 256 bits of entropy.

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800- 133r2. When random values are required, they are obtained from the SP 800-90Ar1 approved DRBG, compliant with Section 4 of SP 800-133r2. The following methods are implemented:

- Key pair generation with ECDSA (CKG): Curves P-224, P-256, P-384, and P-521 with Secret Generation Mode: Testing Candidates.

- Key pair generation with RSA (CKG): Modulus: 2048 and 3072 with Key Generation Mode: B.3.3.

2.10 Key Establishment

N/A

2.11 Industry Protocols

N/A

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Arguments for an API call that provide the data to be used or processed by the module.
N/A	Data Output	Arguments output from an API call.
N/A	Control Input	Arguments for an API call used to control and configure module operation.
N/A	Control Output	N/A
N/A	Status Output	Return values, and/or log messages.

Table 12: Ports and Interfaces

The module's physical perimeter encompasses the case of the tested platform mentioned in Table 2. The module provides its logical interfaces via Application Programming Interface (API) calls. The logical interfaces provided by the module are mapped onto the FIPS 140-3 interfaces (data input, data output, control input, control output and status output) as shown above. The module's data output interface will be disabled when performing the self-test service, zeroization service, or when in an error state.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module supports Crypto Officer (CO) role. The cryptographic module does not provide any authentication methods. The module does not allow concurrent operators. The Crypto Officer is implicitly assumed based on the service requested.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 13: Roles

4.3 Approved Services

The following tables detail the types of approved services available to each role in approved mode of operation, the types of access for each role and the Keys or SSPs they affect.

- Generate G
- Read Access R
- Write Access W
- Execute Access E
- Zeroize Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric encryption	Encrypt a plaintext	Return code “1” denotes use of approved security service	Key, plaintext, initialization vector	Ciphertext	Symmetric encryption Authenticated encryption	Crypto Officer - AES key: W,E
Symmetric decryption	Decrypt a ciphertext	Return code “1” denotes use of approved security service	Key, ciphertext, initialization vector	Plaintext	Symmetric decryption Authenticated decryption	Crypto Officer - AES key: W,E
Key wrapping	Wrap a key	Return code “1” denotes use of approved security service	Key, key to be wrapped	Wrapped key	AES key wrapping	Crypto Officer - AES key: W,E
Key unwrapping	Unwrap a key	Return code “1” denotes use of approved security service	Key, key to be unwrapped	Unwrapped key	AES key unwrapping	Crypto Officer - AES key: W,E
Key derivation	Perform key derivation	Return code “1” denotes use of approved security service	Key derivation key	Derived key	Key derivation with KBKDF	Crypto Officer - Key derivation key: W,E - Derived key: G,R
Message digest generation	Generate message digest	Return code “1” denotes use of approved security service	Message	Message digest	Message digest generation	Crypto Officer
MAC generation	Generate message authentication code	Return code “1” denotes use of approved security service	Message, key	MAC	HMAC Message Authentication Code CMAC Message Authentication Code	Crypto Officer - HMAC key: W,E - CMAC key: W,E
Asymmetric key generation	Asymmetric key generation	Return code “1” denotes use of approved security service	RSA: Padding Method, Modulus size; ECDSA: Curve Type	Key pair	RSA key pair generation ECDSA key pair generation/verification	Crypto Officer - RSA private key: G,R - RSA public key: G,R - ECDSA private key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ECDSA public key: G,R
Digital signature generation	Digital signature generation	Return code “1” denotes use of approved security service	Private key, Message Digest	Signature	RSA signature generation ECDSA signature generation	Crypto Officer - RSA private key: W,E - ECDSA private key: W,E
Digital signature verification	Digital signature verification	Return code “1” denotes use of approved security service	Public key, Message Digest, Signature, RSA: Padding Method, Modulus; ECDAS: Curve Type	Verification result	RSA signature verification ECDSA signature verification	Crypto Officer - RSA public key: W,E - ECDSA public key: W,E
Random number generation	Generate random number	Return code “1” denotes use of approved security service	Number of bytes	Random bytes	Random number generation with Counter DRBG	Crypto Officer - Entropy input string: W,E - DRBG seed: G,E - DRBG internal state V value: G,E - DRBG key: G,E
RSA Decryption primitive	Decryption with RSADP	Return code “1” denotes use of approved security service	RSA private key, Cipher text	Primitive	RSADP primitive	Crypto Officer - RSADP Primitive Private Key: W,E - RSADP Primitive public key: W,E
Show status	Show status of the module	N/A	None	Status information	None	Crypto Officer
Show module's versioning information	Show the versioning information of the module	N/A	None	Module name and version	None	Crypto Officer
Zeroization	Zeroize SSP	N/A	None	None	None	Crypto Officer - AES key: Z - HMAC key: Z - CMAC key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- RSA private key: Z - RSA public key: Z - ECDSA private key: Z - ECDSA public key: Z
Cryptographic algorithm self-test and integrity test	Initiate cryptographic algorithm self-test and integrity test	The successful completion of a service is an implicit indicator for the use of an approved service	N/A	Status information	None	Crypto Officer
Module installation and configuration	Run cryptographic algorithm selftest and integrity test at the module start-up	N/A	N/A	N/A	None	Crypto Officer

Table 14: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
DSA Key Generation	DSA Key Generation [FIPS186-4]	DSA Key Generation [FIPS186-4]	Crypto Officer
DSA Signature Generation	DSA Signature Generation [FIPS186-4]	DSA Signature Generation [FIPS186-4]	Crypto Officer
DSA Signature Verification	DSA Signature Verification [FIPS186-4]	DSA Signature Verification [FIPS186-4]	Crypto Officer
Diffie-Hellman Key Agreement primitive	Diffie-Hellman Key Agreement primitive [SP800-56A Rev 3]	KAS-FFC-SSC [SP800-56A Rev 3]	Crypto Officer
EC Diffie-Hellman Key Agreement primitive	EC Diffie-Hellman Key Agreement primitive [SP800-56A Rev 3]	KAS-ECC-SSC [SP800-56A Rev 3]	Crypto Officer

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

N/A for this module

4.6 Additional Information

The module supports unauthenticated service. The unauthenticated operator can trigger the self-test service by power-cycling the module.

5 Software/Firmware Security

5.1 Integrity Techniques

The module is provided in the form of binary executable code. To ensure the software security, the module is protected by HMAC-SHA2-256 (HMAC Certs. #A3243) algorithm. The software integrity test key (non-SSP) was preloaded to the module's binary at the factory and used for software integrity test only at the pre-operational self-test. At module's initialization, the integrity of the runtime executable is verified using an HMAC-SHA2-256 digest which is compared to a value computed at build time. If at the load time the MAC does not match the stored, known MAC value, the module enters an Error state with all crypto functionality inhibited.

5.2 Initiate on Demand

Integrity tests are performed as part of the Pre-Operational Self-Tests. It is automatically executed at power-on. It can also be invoked by self-test service or powering-off and reloading the module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

The module operates in a modifiable operational environment per FIPS 140-3 level 1 specifications. The module runs within a commercially available kernel of the general-purpose operating system. The module is executing on the hardware specified in the Table 2.

The operating is restricted to a single operator. Only a single instance of the module is allowed in the Operational environment. The operating environment is non-configurable for the operator. The operational environment provides the capability to separate the module during operation from other functions in the operational environment. Those functions do not obtain information from the module related to the CSPs and do not modify CSPs, PSPs, or the execution flow of the module other than via the interfaces provided by the module itself.

The module does not spawn any processes.

7 Physical Security

N/A for this module.

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Volatile memory	Tested platform's RAM for the lifetime of API call, under the module control. The module does not provide persistent storage of SSP.	Dynamic

Table 16: Storage Areas

Keys are not stored inside the cryptographic module. A pointer to a plaintext key is passed through the algorithm APIs. Intermediate keys stored in the module's memory are immediately replaced with 0s in the memory after use. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the module defined API. The operating system protects memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input	Calling application in the TOEPP	Cryptographic module	Plaintext	Manual	Electronic	
API output	Cryptographic module	Calling application in the TOEPP	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Crypto transformation context destructor	Crypto transformation context destructor zeroizes all SSPs stored in its context struct	SSPs are actively overwritten with zeroes and thus not recoverable	Using the appropriate zeroization function OPENSSL_cleanse()
Cycling the power to the tested platform	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten when power is removed	By cycling power

Table 18: SSP Zeroization Methods

The zeroization mechanism for all of the CSPs is to replace 0s in the memory which originally stored the CSPs. Zeroization of sensitive data is performed automatically by calling zeroization API function OPENSSL_cleanse() for temporarily stored CSPs or cycling the power to the tested platform. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module. Input and output interfaces are inhibited while zeroization is performed.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	Keys used for AES encryption / decryption	128, 192, 256 bits - 128 to 256 bits	Symmetric AES key - CSP			Symmetric encryption Symmetric decryption Authenticated encryption Authenticated decryption
CMAC key	Keys used for CMAC generation	128 bits - 128 bits	Symmetric AES key - CSP			CMAC Message Authentication Code
HMAC key	Keys used for HMAC computation	160 bits or greater - 160 bits or greater	Symmetric HMAC key - CSP			HMAC Message Authentication Code
RSA private key	RSA private key for digital signature computations	Up to 4096 bits - 112 bits or greater	Asymmetric RSA private key - CSP	RSA key pair generation		RSA signature generation
RSA public key	RSA public key for digital signature computations	Up to 4096 bits - Less than 112 bits for module size 1024 bits, greater than 112 bits for modulus sizes of 2048 bits or greater	Asymmetric RSA public key - PSP	RSA key pair generation		RSA signature verification
ECDSA private key	ECDSA private key for digital signature computations	Up to 521 bits - 112 bits or greater	Asymmetric ECDSA private key - CSP	ECDSA key pair generation/verification		ECDSA signature generation
ECDSA public key	ECDSA public key for digital signature computations	Up to 521 bits - 112 bits or greater	Asymmetric ECDSA public key - PSP		ECDSA key pair generation/verification	ECDSA signature verification

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Entropy input string	Entropy input string used to seed the Counter DRBG	6144 bits - at least 256 bits	Entropy input - CSP			Random number generation with Counter DRBG
DRBG seed	Seed of Counter DRBG	384 bits - at least 256 bits	Seed of Counter DRBG - CSP	Random number generation with Counter DRBG		Random number generation with Counter DRBG
DRBG internal state V value	Value V of internal state of Counter DRBG	128 bits - 128 bits	Value V of internal state of Counter DRBG - CSP	Random number generation with Counter DRBG		Random number generation with Counter DRBG
DRBG key	Key of Counter DRBG	256 bits - 256 bits	Key of Counter DRBG - CSP	Random number generation with Counter DRBG		Random number generation with Counter DRBG
Key derivation key	Key for key derivation	256 bits - 256 bits	Key of key derivation - CSP			Key derivation with KBKDF
Derived key	Derived key	512-4096 bits - at least 112 bits	Derived key - CSP	Key derivation with KBKDF		
RSADP Primitive Private Key	Used for RSADP primitive component	2048 bits - 112 bits	Private key - CSP	RSA key pair generation		RSADP primitive
RSADP Primitive public key	Used for RSADP primitive component	2048 bits - 112 bits	Public key - PSP		RSA key pair generation	

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	
CMAC key	API input	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	
HMAC key	API input	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Cycling the power to the tested platform	
RSA private key	API input API output	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	RSA public key:Paired With
RSA public key	API input API output	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	RSA private key:Paired With
ECDSA private key	API input API output	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	ECDSA public key:Paired With
ECDSA public key	API input API output	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	ECDSA private key:Paired With
Entropy input string	API input	Volatile memory:Plaintext	For the lifetime of the API call	Crypto transformation context destructor Cycling the power to the tested platform	DRBG seed:Generates DRBG internal state V value:Generates DRBG key:Generates
DRBG seed		Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	Entropy input string:Derived from DRBG internal state V value:Generates DRBG key:Generates
DRBG internal state V value		Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	DRBG seed:Derived from DRBG key:Paired With
DRBG key		Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	DRBG seed:Derived from DRBG internal state V value:Paired With
Key derivation key	API input	Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	Derived key:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Derived key	API output	Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	Key derivation key:Derived from
RSADP Primitive Private Key		Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	RSADP Primitive public key:Paired With
RSADP Primitive public key		Volatile memory:Plaintext	For the runtime of the module or until zeroization	Crypto transformation context destructor Cycling the power to the tested platform	RSADP Primitive Private Key:Paired With

Table 20: SSP Table 2

9.5 Transitions

SHA-1
The module includes an implementation of SHA-1 for hashing and digital signature verification. This implementation will be non-Approved for all uses starting January 1, 2031. At this time, the user should move to SHA2, which is available in this module.

186-4/186-5
As of February 5, 2024, the CMVP does not accept module submissions that implement DSA or RSA X9.31 in the approved mode, other than for signature verification which is approved for legacy use. This module does not implement DSA or RSA X9.31 for signature generation and therefore is unaffected by the current transition from 186-4 to 186-5. As detailed in section 2.7, the CAVP testing performed on the 186-4 algorithms is mathematically similar to the testing performed on the 186-5 algorithms and therefore this module claims compliance with 186-5. This means that no timeline exists in which any of the implemented algorithms will transition from approved to non-approved."

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3243)	HMAC-SHA2-256	KAT	SW/FW Integrity	Module is in normal state	Module performs HMAC-SHA2-256 KAT prior to firmware integrity test

Table 21: Pre-Operational Self-Tests

The module performs Pre-operational Self-tests automatically when the module is loaded into memory (i.e. at power on). The Pre-operational Self-tests contain pre-operational software integrity test to ensure that the module is not corrupted. The integrity test is performed on the runtime image of the module using HMAC-SHA2-256. Prior to software integrity test, a CAST for HMAC-SHA2-256 is performed. If the CAST on the HMAC-SHA-256 is successful, the HMAC value of the runtime image is recalculated and compared with the stored HMAC value pre-computed at compilation time (for details, see also Section 5 Software/Firmware Security). While the module is performing the Pre-operational Self-tests no other functions are available and all output is inhibited. Once Pre-operational Self-tests are completed successfully, the module enters operational mode and cryptographic services are available.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB Encrypt/Decrypt KAT (A3243)	Key lengths: 128 bits	KAT	CAST	Module is in normal state	Encryption, Decryption	During module start-up
AES-CMAC Encrypt/Decrypt KAT (A3243)	Key lengths: 128, 256 bits	KAT	CAST	Module is in normal state	MAC Generation	During the module start-up
AES-GCM Authenticated Encrypt/Decrypt KAT (A3243)	Key lengths: 256 bits	KAT	CAST	Module is in normal state	Encryption, Decryption	During the module start-up
AES-KW Encrypt/Decrypt KAT (A3243)	Key lengths: 256 bits	KAT	CAST	Module is in normal state	Encryption, Decryption	During the module start-up
Counter DRBG Instantiate/Generate/Reseed KAT (A3243)	AES-256	KAT	CAST	Module is in normal state	Instantiate, Generate, Reseed	During module start-up
ECDSA SigGen KAT (A3243)	P-256 with SHA2-256	KAT	CAST	Module is in normal state	Signature Generation	During module start-up
ECDSA SigVer KAT (A3243)	P-256 with SHA2-256	KAT	CAST	Module is in normal state	Signature Verification	During module start-up
HMAC-SHA2-256 KAT (A3243)	SHA2-256	KAT	CAST	Module is in normal state	MAC Generation	During module start-up
RSA SigGen KAT (A3243)	2048 bits with SHA2-256	KAT	CAST	Module is in normal state	Signature Generation	During module start-up
RSA SigVer KAT (A3243)	2048 bits with SHA2-256	KAT	CAST	Module is in normal state	Signature Verification	During module start-up
SHA1 KAT (A3243)	N/A	KAT	CAST	Module is in normal state	Hash Generation	During module start-up
SHA2-256 KAT (A3243)	N/A	KAT	CAST	Module is in normal state	Hash Generation	During module start-up
SHA2-512 KAT (A3243)	N/A	KAT	CAST	Module is in normal state	Hash Generation	During module start-up
KDF SP800-108 KAT (A3243)	HMAC-SHA2-512	KAT	CAST	When the test passes, FIPS_status()	Key Derivation	During module start-up

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				returns 1, otherwise it returns 0		

Table 22: Conditional Self-Tests

The module performs conditional cryptographic algorithm self-tests (CASTs) at module initialization to ensure that the algorithms work as expected, before any security function or process is invoked via module interface.
The module performs self-tests that cover all Approved cryptographic algorithms supported in the approved mode of operation using the Known-answer Tests (KAT) as shown in the table below. None of the keys used for the KAT are considered as SSP.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3243)	KAT	SW/FW Integrity	On Demand	Reloading the module or using SCRYPTO_post()

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB Encrypt/Decrypt KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
AES-CMAC Encrypt/Decrypt KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post(1)
AES-GCM Authenticated Encrypt/Decrypt KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post(1)
AES-KW Encrypt/Decrypt KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
Counter DRBG Instantiate/Generate/Reseed KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
ECDSA SigGen KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
ECDSA SigVer KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
HMAC-SHA2-256 KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
RSA SigGen KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
RSA SigVer KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
SHA1 KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256 KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
SHA2-512 KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()
KDF SP800-108 KAT (A3243)	KAT	CAST	On demand	Reloading the module or using SCRYPTO_post()

Table 24: Conditional Periodic Information

The module provides the service to perform both Pre-operational self-test and CASTs on-demand by calling SCRYPTO_post() function. This service performs all the cryptographic algorithm tests listed in the Conditional Self-Tests table and pre-operational software integrity test. During the execution of the on-demand self-tests, no other functions are available and all output is inhibited. If any of the tests fail, the module will enter the Error state.

10.4 Operator Initiation of Self-Tests

On demand and periodic self-tests are performed by powering off the module and powering it on again. This service performs the same cryptographic algorithm tests executed during pre-operational self-tests and CASTs. During the execution of the periodic and on-demand self-tests, crypto services are not available and no data output or input is possible.

10.5 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The module's only error state	Any failure in context of the execution of the implemented self-tests during module start-up or the self-test service	Reloading the module	The OE`s OS log contains message: <ERROR> 'Algorithm name' Selftest failed. <ERROR> Algorithm Self-test Failed.

Table 25: Error States

The module has an API indicating the status of the Self-test FIPS_status(). It returns 1 while the module is in the operational state, otherwise 0 while the modules is in the Error state. In the Error state, no cryptographic services are provided, and data output is prohibited.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is built into the operational environment and delivered with a device. There is no standalone delivery of the module as a software library.

Secure initialization and startup
The module is initialized during the loading of the module before any cryptographic functionality is available. The operating system is responsible for the initialization and loading processes of the module. The module is designed with constructor (default entry point of the module) which

ensures that the cryptographic algorithm self-tests (CASTs) and pre-operational self-test are initiated automatically when the module is loaded.

Secure operation

The module is provided directly to solution developers and is not available for direct download to the general public. The module is installed on an operating system specified in Section 2.1. Additional Rules of Operation:

1. The writable memory areas of the module (data and stack segments) are accessible only by the application so that the operating system is in "single user" mode, i.e. only the application has access to that instance of the module.

2. The operating system is responsible for multiprocessing operations so that other processes cannot access the address space of the process containing the module.

3. Only the services defined in Table 8 shall be used in Approved Mode of operation.

11.2 Administrator Guidance

No specific Administrator guidance.

11.3 Non-Administrator Guidance

No specific Non-Administrator guidance.

12 Mitigation of Other Attacks

The module does not implement security mechanisms to mitigate other attacks.