Samsung Electronics Co., Ltd.

Samsung SCrypto Cryptographic Module

Software Version: 2.7

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.2

Last Update: 8-30-2024

Contents

Copyrights and Trademarks

1. General

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.

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 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

Table 1 - Security Levels

Purpose of the Security Policy

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.

Target Audience

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

The following section describes the cryptographic module and how it conforms to the FIPS 140-3 specification in each of the required areas.

Module overview

The 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.

The module has been tested on the following platforms:

#	Operating System	Hardware Platform	Processor	PAA/Acceleration
1	QSEE 5.24 (64-bit)	Samsung Galaxy S23+	Qualcomm Snapdragon 8 Gen 2	Not implemented

Table 2 - Tested Operational Environments

Note: The 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.

The "Vendor Affirmed Operational Environments" table defined in SP 800-140B is missing because no other operational environment is vendor affirmed.

Modes of operation

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. See Table 3 and Table 8 below for a list of the supported approved/allowed algorithms/services. The non-approved mode is entered when a non-approved algorithm/non-approved service is invoked. See Table 5 and Table 9 for a list of non-approved algorithms/non-approved services. When the module is initialized, the self-tests are executed automatically. After successful completion of self-test, the module enters operational state.

Module supports only normal operation. Degraded operation is not supported.

The following table shows the Approved algorithms that can be used in Approved Mode of Operation:

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A3243	AES [FIPS 197] [SP 800-38A]	AES-ECB	Key Length: 128, 192, 256 bits	Symmetric Encryption and Decryption
#A3243	AES [FIPS 197] [SP 800-38A]	AES-CBC	Key Length: 128, 192, 256 bits	Symmetric Encryption and Decryption
#A3243	AES [FIPS 197] [SP 800-38A]	AES-CTR	Key Length: 128, 192, 256 bits	Symmetric Encryption and Decryption
#A3243	AES [FIPS 197] [SP 800-38A]	AES-OFB	Key Length: 128, 256 bits	Symmetric Encryption and Decryption
#A3243	AES [FIPS 197] [SP 800-38B]	AES-CMAC	Key Length: 128, 192, 256 bits	Message Authentication

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
#A3243	AES [FIPS 197] [SP 800-38D]	AES-GCM	Key Length: 128, 192, 256 bits	Authenticated Symmetric Encryption and Decryption
#A3243	AES [FIPS 197] [SP 800-38F]	AES-KW	Key Length: 128, 192, 256 bits	Key Wrapping and Unwrapping
#A3243	ECDSA [FIPS 186-4]	KeyGen	Curve: P-224, P-256, P-384, P-521	Asymmetric Key Generation
#A3243	ECDSA [FIPS 186-4]	KeyVer	Curve: P-224, P-256, P-384, P-521	Asymmetric Public Key Verification
#A3243	ECDSA [FIPS 186-4]	SigGen	Curve: P-224, P-256, P-384, P-521	Digital Signature Generation
#A3243	ECDSA [FIPS 186-4]	SigVer	Curve: P-224, P-256, P-384, P-521	Digital Signature Verification
#A3243	DRBG [SP800-90Arev1]	CTR_DRBG with AES-256 Derivation Function Disabled No Prediction Resistance	Key Length: 256 bits	Random Number Generation
#A3243	HMAC [FIPS 198-1]	SHA-1	Key Length 112 bits or greater	Keyed Hash
#A3243	HMAC [FIPS 198-1]	SHA2-224	Key Length 112 bits or greater	Keyed Hash
#A3243	HMAC [FIPS 198-1]	SHA2-256	Key Length 112 bits or greater	Keyed Hash
#A3243	HMAC [FIPS 198-1]	SHA2-2384	Key Length 112 bits or greater	Keyed Hash
#A3243	HMAC [FIPS 198-1]	SHA2-512	Key Length 112 bits or greater	Keyed Hash
#A3243	SHS [FIPS 180-4]	SHA-1	N/A	Message Digest Note: SHA-1 is not used for digital signature generation
#A3243	SHS [FIPS 180-4]	SHA2-224	N/A	Message Digest
#A3243	SHS [FIPS 180-4]	SHA2-256	N/A	Message Digest
#A3243	SHS [FIPS 180-4]	SHA2-384	N/A	Message Digest
#A3243	SHS [FIPS 180-4]	SHA2-512	N/A	Message Digest
#A3243	RSA [FIPS 186-4]	Key Generation Mode: B.3.3, Primality Tests: C.2	Modulus: 2048, 3072	Asymmetric Key Generation
#A3243	RSA [FIPS 186-4]	Signature Generation (PKCS#1 v1.5) and (PKCS-PSS)	Modulus: 2048, 3072	Digital Signature Generation
#A3243	RSA [FIPS 186-4]	Signature Verification (PKCS#1 v1.5) and (PKCS-PSS)	Modulus: 1024, 2048, 3072	Digital Signature Verification
#A3243	KBKDF [SP800- 108] (CVL)	KDF Mode: counter MAC Mode: HMAC-SHA2-512	Supported Length: 512-4096 Increment 1 Fixed Data Order: after/before/middle fixed data Counter Length: 8, 16, 24, 32	Key Derivation

#A3243	RSA Decryption Primitive [SP800-56Brev2] (CVL)	N/A	Modulus: 2048	RSADP Decryption
Vendor Affirmed	CKG (SP800-133rev2)	Section 5.1, Section 5.2	Cryptographic Key Generation; SP 800-133rev2 and IG D.H.	Key generation. Note: The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per section 5 in SP800-133rev2 (vendor affirmed). A seed (i.e., the random value) used in asymmetric key generation is a direct output from SP800-90Arev1 CTR_DRBG

Table 3 - Approved Algorithms

Notes:

- There are some algorithm modes that were tested but not implemented by the module. Only the algorithms, modes, and key sizes that are implemented by the module are shown in Table 3 above.

- 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.

Vendor Name	Certificate Number
Qualcomm Technologies, Inc.	E67

Table 4 – Entropy Certificate

In addition to the above listed Approved/Allowed services, the cryptographic module also provides non-Approved services; however, any use of the module's non-Approved services causes the module to operate in a non-approved manner. Thus, operators shall not utilize any of the following non-Approved service(s).

Algorithm/Function	Use/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 5 – Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

Please note that due to the lack of the associated self-tests to DSA, KAS-ECC-SSC and KAS-FFC-SSC algorithms, Table 5 lists those algorithms as the Non-Approved Algorithms Not Allowed in the Approved Mode of Operation.

The "Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed" table defined in SP 800-140B is missing because the module does not implement any such algorithms.

The "Non-Approved Algorithms Allowed in the Approved Mode of Operation" table defined in SP 800-140B is missing because the module does not implement any such algorithms.

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.

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 file *scrypto_v2.7_x64_qsee_release.a* for the tested platform running QSEE 5.24 (64-bit).

Figure 1 below illustrates a block diagram of a typical GPC and the module's physical perimeter. The module's cryptographic boundary consists of all functionalities contained within the module's compiled source code.

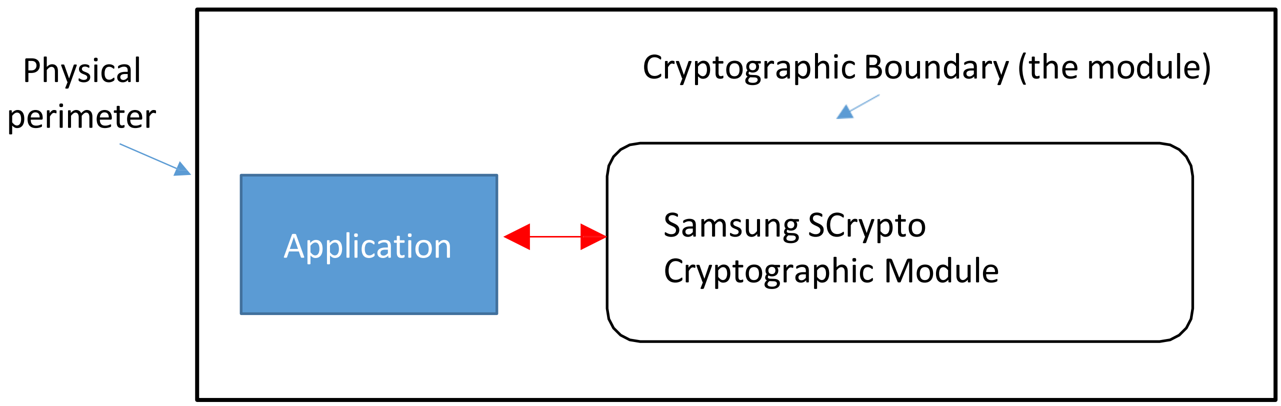

Figure 1 - Logical block diagram

3. Cryptographic module interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-3 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs.

The module does not implement a trusted channel.

The logical interfaces are the application program interface (API) through which applications request services. The following table summarizes the logical interfaces.

Physical port	Logical interface	Data that passes over port/interface
N/A	Data input interface	Arguments for an API call that provide the data to be used or processed by the module
N/A	Data output interface	Arguments output from an API call
N/A	Control input interface	Arguments for an API call used to control and configure module operation
N/A	Control output interface	Not applicable
N/A	Status output interface	Return values, and or log messages

Table 6 – Ports and Interfaces

4. Roles, services, and authentication

The module supports the single role of **Crypto Officer (CO)**, which performs all services including module installation and configuration.

The Crypto Officer role is implicitly assumed by the entity accessing the module services.

The module does not support user authentication.

The module does not implement a bypass capability.

The module does not implement a self-initiated cryptographic output capability.

The module does not support Software loading.

Role	Service	Input	Output
CO	Symmetric encryption/decryption	Input for Encryption: key and plain text Input for Decryption: key and cipher text	Output for Encryption: cipher text; Output for Decryption: plain text
CO	Asymmetric key generation	RSA - Padding Method, Modulo size, ECDSA - Curve Type	Key pair
CO	Key wrapping	Key	Wrapped key
CO	Digital signature generation	Private key, Message Digest	Signature
CO	Digital signature verification	Public key, Message Digest, Signature, RSA - Padding Method, Modulo n, ECDSA - Curve Type	Verification result
CO	Message digest generation	Message	Message digest
CO	MAC generation	Key, message	Message authentication code
CO	Random Number Generation	Entropy input string, Personalization string, Additional input	Random bits
CO	Key derivation	Key (SP800-108)	Derived Key
CO	RSA Decryption primitive	RSA private key, Cipher text	Message
CO	Show status	None	Module's status
CO	Show version	None	Module's name/ID and versioning information
CO	Zeroization	SSPs	Zeroized and released memory space
CO	Cryptographic Algorithm Self-Test and Integrity Test	None	Self-test status
CO	Module Installation and Configuration	None	None

Table 7 – Roles, Service Commands, Input and Output

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Symmetric encryption/ decryption	Encrypt a plain text or Decrypt a cipher text	AES-ECB; AES-CBC; AES-OFB; AES-CTR; AES-GCM;	AES key	CO	W, E	Return code “1” denotes use of approved security service

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Asymmetric key generation	Generate asymmetric key pair	CKG; CTR_DRBG; RSA KeyGen; ECDSA KeyGen; ECDSA KeyVer;	RSA private key; RSA public key; ECDSA private key; ECDSA public key;	CO	G, R, W	Return code "1" denotes use of approved security service
Key wrapping	Encrypt or decrypt a key value	AES-KW	AES key wrapping key	CO	W, E	Return code "1" denotes use of approved security service
Digital signature generation	Generate digital signature	RSA SigGen; ECDSA SigGen;	RSA private key; ECDSA private key;	CO	W, E	Return code "1" denotes use of approved security service
Digital signature verification	Verify digital signature	RSA SigVer; ECDSA SigVer;	RSA public key; ECDSA public key;	CO	W, E	Return code "1" denotes use of approved security service
Message digest generation	Generate message digest	SHA-1; SHA2-224; SHA2-256; SHA2- 384; SHA2-512;	None	CO	N/A	Return code "1" denotes use of approved security service
MAC generation	Generate message authentication code	AES-CMAC; HMAC-SHA-1; HMAC-SHA2-224; HMAC-SHA2-256; HMAC-SHA2-384; HMAC-SHA2-512;	HMAC key; CMAC key;	CO	W, E	Return code "1" denotes use of approved security service
Random Number Generation	Generate random number	CTR_DRBG	Entropy input string; DRBG seed; DRBG internal state V value; DRBG key;	CO	G, R, E	Return code "1" denotes use of approved security service
Key derivation	Derive keying material	KBKDF	KBKDF key-derivation key	CO	W, E	Return code "1" denotes use of approved security service
RSA Decryption primitive	Decryption with RSADP	RSA Decryption Primitive	RSA private key	CO	W, E	Return code "1" denotes use of approved security service
Show status	Provide Module's current status (status message)	N/A	N/A	CO	N/A	N/A
Show version	Provide Module's name and version information	N/A	N/A	CO	N/A	N/A
Zeroization	Zeroize SSP	N/A	ALL SSPs	CO	Z	N/A

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access rights to Keys and/or SSPs	Indicator
Cryptographic Algorithm Self-Test and Integrity Test	Initiate cryptographic algorithm self-test and integrity test	AES-ECB; AES-CMAC; AES-GCM; AES-KW; DRBG; ECDSA Sign; ECDSA verify; HMAC-SHA2-256; KBKDF; RSA Sign; RSA Verify; SHA-1; SHA2-256; SHA2-512;	N/A	CO	N/A	N/A
Module Installation and Configuration	Run cryptographic algorithm self-test and integrity test at the module start-up	N/A	N/A	CO	N/A	N/A

Table 8 – Approved Services

Service	Description and Input/Output	Algorithms Accessed	Roles	Indicator
DSA Key Generation	DSA Key Generation [FIPS186-4]	DSA	CO	Return code “0” denotes use of non-approved security service
DSA Signature Generation	DSA Signature Generation [FIPS186-4]	DSA	CO	Return code “0” denotes use of non-approved security service
DSA Signature Verification	DSA Signature Verification [FIPS186-4]	DSA	CO	Return code “0” denotes use of non-approved security service
Diffie-Hellman Key Agreement primitive	Diffie-Hellman Key Agreement primitive [SP800-56A Rev 3]	KAS-FFC-SSC	CO	Return code “0” denotes use of non-approved security service
EC Diffie-Hellman Key Agreement primitive	EC Diffie-Hellman Key Agreement primitive [SP800-56A Rev 3]	KAS-ECC-SSC	CO	Return code “0” denotes use of non-approved security service

Table 9 – Non-Approved Services

G = Generate: The module generates or derives the SSP.

R = Read: The SSP is read from the module (e.g. the SSP is output).

W = Write: The SSP is updated, imported, or written to the module.

E = Execute: The module uses the SSP in performing a cryptographic operation.

Z = Zeroise: The module zeroises the SSP.

The approved security service indicator of the module is compliant to the example scenario 2) of [IG] 2.4.C.

5. Software/Firmware security

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 would enter an Error state with all crypto functionality inhibited.

On-Demand Integrity Test

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

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

The operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded). The external application that makes calls to the cryptographic module is the single user of the module, even when the application is serving multiple clients. 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.

7. Physical security

The module is comprised of software only and thus does not claim any physical security.

8. Non-invasive security

The module does not implement non-invasive attack mitigation techniques to protect the module's unprotected SSPs from non-invasive attacks referenced in Annex F of FIPS 140-3.

9. Sensitive security parameter management

Key/SSP Name/Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establi-shment	Storage	Zeroisation	Use & related keys
AES keys (CSP)	128, 192 and 256 bits	AES-ECB, AES-CBC; AES-OFB; AES-CTR; AES-GCM Algo Cert. #A3243	N/A	Import from calling application within TOEPP No Export	None	Tested platform's RAM for the lifetime of API call, under the module control Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Symmetric Encryption / Decryption
AES key wrapping key (CSP)	128, 192 and 256 bits	AES-KW Algo Cert. #A3243	N/A	Import from Calling application within TOEPP No Export	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Key wrapping and unwrapping
CMAC keys (CSP)	128, 192 and 256 bits	AES-CMAC Algo Cert. #A3243	N/A	Import from calling application within TOEPP No Export	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	CMAC Generation
HMAC keys (CSP)	Min 112 bits	HMAC-SHA-1; HMAC-SHA2-224; HMAC-SHA2-256; HMAC-SHA2-384; HMAC-SHA2-512; Algo Cert. #A3243	N/A	Import from calling application within TOEPP No Export	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Keyed Hash

Key/SSP Name/Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establi-shment	Storage	Zeroisation	Use & related keys
RSA private key (CSP)	Equal to 2048-bit, 3072-bit RSA key	DRBG; RSA KeyGen; RSA SigGen Algo Cert. #A3243	Internally generated conformant to SP800-133r2 (CKG) using FIPS 186-4 RSA key generation method, and the random value used in the key generation is generated using SP800-90Arev1 DRBG	Import and Export to Calling application within TOEPP.	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Digital Signature Generation Related: RSA public key
RSA public key (PSP)	Equal to 2048-bit, 3072-bit RSA key	RSA SigVer Algo Cert. #A3243	Internally derived per the FIPS 186-4 RSA key generation method	Import and Export to Calling application within TOEPP.	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Digital Signature Verification Related: RSA private key
ECDSA private key (CSP)	Equal to 224-bit, 256-bit, 384-bit, 521-bit ECC key	DRBG; ECDSA KeyGen; ECDSA KeyVer; ECDSA SigGen; Algo Cert. #A3243	Internally generated conformant to SP800-133r2 (CKG) using FIPS 186-4 ECDSA key generation method, and the random value used in the key generation is generated using SP800-90Arev1 DRBG	Import and Export to Calling application within TOEPP.	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Digital Signature Generation Related: ECDSA public key
ECDSA public key (PSP)	Equal to 224-bit, 256-bit, 384-bit, 521-bit ECC key	ECDSA SigVer Algo Cert. #A3243	Internally derived per the FIPS 186-4 ECDSA key generation method	Import and Export to calling application within TOEPP.	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Digital Signature Verification Related: ECDSA private key

Key/SSP Name/Type	Strength	Security Function and Cert. Number	Generation	Import / Export	Establi-shment	Storage	Zeroisation	Use & related keys
KBKDF key-derivation key (CSP)	At least 112 bits	KBKDF Algo Cert. #A3243	N/A	Import from calling application within TOEPP No Export	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage.	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Key Derivation
Entropy input string (CSP)	384 bits	CTR_DRBG Algo Cert. #A3243	Obtained from the Entropy Source within TOEPP	Import to the module via Module's API within TOEPP Export: No	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Random Number Generation
DRBG seed (CSP)	256 bits	CTR_DRBG Algo Cert. #A3243	Internally Derived from entropy input string as defined by SP800-90Arev1	N/A	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Random Number Generation
DRBG internal s tate V value (CSP)	256 bits	CTR_DRBG Algo Cert. #A3243	Internally Derived from entropy input string as defined by SP800-90Arev1	N/A	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Random Number Generation
DRBG key (CSP)	256 bits	CTR_DRBG Algo Cert. #A3243	Internally Derived from entropy input string as defined by SP800-90Arev1	N/A	None	Tested platform's RAM for the lifetime of API call, under the module control. Note: The module does not provide persistent keys/ SSPs storage	By calling OPENSSL_cleanse function or cycling the power to the tested platform	Random Number Generation

Table 10 – SSPs

Entropy sources	Minimum number of bits of entropy	Details
Snapdragon(R) 8 Gen 2 Mobile Platform developed by Qualcomm Technologies, Inc. Implementation Name: Entropy Source of the Qualcomm(R) Pseudo Random Number Generator	0.420625 bits/sample (4 bits)	ESV Cert. #E67 Physical Entropy Source. Used to seed approved SP800-90A DRBG. The entropy source is located inside the module's physical perimeter, but the outside the module's boundary

Table 11 – Non-Deterministic Random Number Generation Specification

Random number generation

The module employs an Approved SP 800-90Arev1 CTR_DRBG for creation of random numbers. The module uses the physical entropy source (ESV Cert. #E67) from the operational environment as the source of random numbers for DRBG seeds. 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.

Use of RBG output

The module's SP800-90Arev1 CTR_DRBG is used to generate random numbers for key generation.

The calling application is responsible for storage of generated keys returned by the module. It is not possible for the module to output information during the key generating process.

SSP entry and output

All keys and SSPs that are entered from or output to the module are entered from or output to the invoking application running on the same device. Keys/SSPs entered into the module are electronically entered in plain text form. Keys/SSPs are output from the module in plain text form if required by the calling application. The module does not support manual key entry or key output. Keys or other CSPs can only be exchanged between the module and the calling application using appropriate API calls. The module does not output intermediate key generation values.

The module performs two independent internal actions for the output of plaintext CSPs:

1. The module calls the random number generator service and verifies that the service has been completed without any errors.
2. The module performs the Pair-wise Consistency test and verifies that the test is completed without any errors.

Only after the successful completion of these two actions will the module allow the output of plaintext CSPs.

SSP storage

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.

SSP zeroization

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.

10. Self-tests

The module performs a series of power-up self-tests, that covers all of its approved algorithms. The module executes all self-tests when the module is initialized during the boot process. Self-tests can also be manually invoked by calling FIPS_SCRYPTO_post(1). When the module passes all of its power-up Self-tests, the module sets an internal variable to reflect this. A calling application can call the FIPS_status() API to obtain the value of this internal variable (1 if the Self-test was successful, and 0 otherwise if the Self-test failed). In addition to Known Answer Tests (KATs) for each of the module's cryptographic algorithms, the module also performs a binary integrity test to check for corruption. If any KAT self-test or the integrity test fails, the module sets its error flag (static variable), returns an error code to the API function caller to indicate the error, enters an error state (FIPS_ERR), and inhibits Crypto APIs that return cryptographic information. While the module is executing the self-tests, services are not available, and input and output are inhibited.

Pre-operational self-test

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). 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.

Conditional self-tests

Conditional cryptographic algorithm 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.

Algorithm	Test	Condition
AES	AES-ECB with 128 bits Encryption KAT AES-ECB with 128 bits Decryption KAT	Start-up, on-demand
AES-CMAC	AES-CMAC with 128 bits MAC Generation KAT AES-CMAC with 256 bits MAC Generation KAT	Start-up, on-demand
AES-GCM	AES-GCM with 256 bits Authenticated Encryption KAT AES-GCM with 256 bits Authenticated Decryption KAT	Start-up, on-demand
AES-KW	AES-KW with 256 bits Encryption KAT AES-KW with 256 bits Decryption KAT	Start-up, on-demand
SHA	SHA-1 KAT SHA2-256 KAT SHA2-512 KAT	Start-up, on-demand
ECDSA	ECDSA P-256 with SHA2-256 SigGen KAT	Start-up, on-demand
ECDSA	ECDSA P-256 with SHA2-256 SigVer KAT	Start-up, on-demand
SP800-108 KDF	KBKDF KAT	Start-up, on-demand
RSA	RSA 2048 modulus with SHA2-256 SigGen KAT	Start-up, on-demand
RSA	RSA 2048 modulus with SHA2-256 SigVer KAT	Start-up, on-demand
HMAC	HMAC-SHA2-256 KAT	Start-up, on-demand
DRBG	CTR_DRBG Instantiate KAT CTR_DRBG Generate KAT CTR_DRBG Reseed KAT	

	Note: DRBG Health Tests as specified in NIST SP 800-90Arev1 Section 11.3 are performed)	

Table 13 – Self-Tests

The Entropy Source is outside of the module cryptographic boundary, but it is within the boundary of the TOEPP, and the module itself does not perform Entropy Source Health Tests.

Algorithm	Test	Condition
SP800-90B Entropy Source	Repetition Count Test (RCT) and Adaptive Proportion Test (APT)	Start-up, Continuous and on-demand

Table 14 – Entropy Source Health Tests

Conditional pair-wise consistency tests

The module performs Pair-wise Consistency Tests (PCT) on the cryptographic algorithms shown in the following table. If any of the PCT fail, the module enters the Error state.

Algorithm	Test	Condition
ECDSA	Pair-wise consistency test	After key pair generation prior to the first exportation, or prior to the first operational use
RSA	Pair-wise consistency test	After key pair generation prior to the first exportation, or prior to the first operational use

Table 15 – Pair-wise consistency tests

Periodic self-tests

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 Table 13 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.

Error state and status indicators

The module has an API indicating the status of the Self-test FIPS_status(). It returns 1 while the modules 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

Secure installation

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.

Maintenance requirements

The module does not support maintenance role.

End of life

The module does not provide persistent storage for keys, SSPs, user data, etc. The module does not store any sensitive information beyond the lifetime of an API call. Intermediate CSPs stored in the memory of the module are immediately replaced with 0s in the memory after use. The end user of the operating system is also responsible for zeroizing SSPs when the cryptographic module is no longer deployed or intended for further use by the operator.

12. Mitigation of other attacks

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

Glossary and Abbreviations

AES	Advanced Encryption Specification
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter mode of AES
CVL	Component Validation List
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards Publication
HMAC	Hash Message Authentication Code
KAT	Known-answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
POST	Pre-Operational Self-Test
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

References

FIPS180-4 **Secure Hash Standard (SHS)**
August 2015
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf

IG **Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program**
October, 2022
https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**
Specifications Version 2.1
February 2003
https://www.ietf.org/rfc/rfc3447.txt

SP800-38A **NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
https://csrc.nist.gov/publications/detail/sp/800-38a/final

SP800-38B **NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**
May 2005
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf

SP800-56Ar3 **NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography**
April 2018
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

SP800-90Ar1 **NIST Special Publication 800-90A Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
June 2015
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

SP800-133r2 **NIST Special Publication 800-133 Revision 2 - Recommendation for Cryptographic Key Generation**
June 2020
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf