SAMSUNG

Samsung Electronics Co., Ltd.

Samsung Kernel Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

List of Tables

List of Figures

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

1 General

1.1 Overview

This document is a non-proprietary FIPS 140-3 Security Policy for the Samsung Kernel 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

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Samsung Kernel Cryptographic Module (hereafter referred to as “the module” or SKC) is a software module running on a multi-chip standalone general-purpose computing platform. The module’s software version number is 2.3. The module provides cryptographic services to Kernel through an application program interface (API). The binary image that contains the Samsung Kernel Cryptographic Module for the appropriate platform is boot.img.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Module Characteristics:

Cryptographic Boundary:

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

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. 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 and comprises the following object files components. The object files are generated from the sources through the kernel build process. The module is intended only for single-process execution.

#	Object file name
1	fips140_integrity.o
2	fips140_post.o
3	fips140_test.o
4	fips140_out.o
5	fips140_3_services.o
6	api.o
7	cipher.o
8	algapi.o
9	scatterwalk.o
10	skcipher.o
11	ahash.o
12	shash.o
13	hmac.o
14	sha1_generic.o
15	sha256_generic.o
16	sha512_generic.o
17	ecb.o
18	cbc.o
19	aes_generic.o
20	aes-ce-core.o
21	aes-ce-glue.o
22	aes-ce.o
23	aes-glue-ce.o
24	sha256-core.o
25	sha256-glue.o
26	sha2-ce-core.o
27	sha2-ce-glue.o
28	sha1-ce-core.o
29	sha1-ce-glue.o

Tested Operational Environment's Physical Perimeter (TOEPP):

The boundary of the TOEPP is defined as the entire chassis unit's physical perimeter encompassing the "top," "front," "left," "right," "rear" and "bottom" surfaces of the case, and shown in the figures below.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

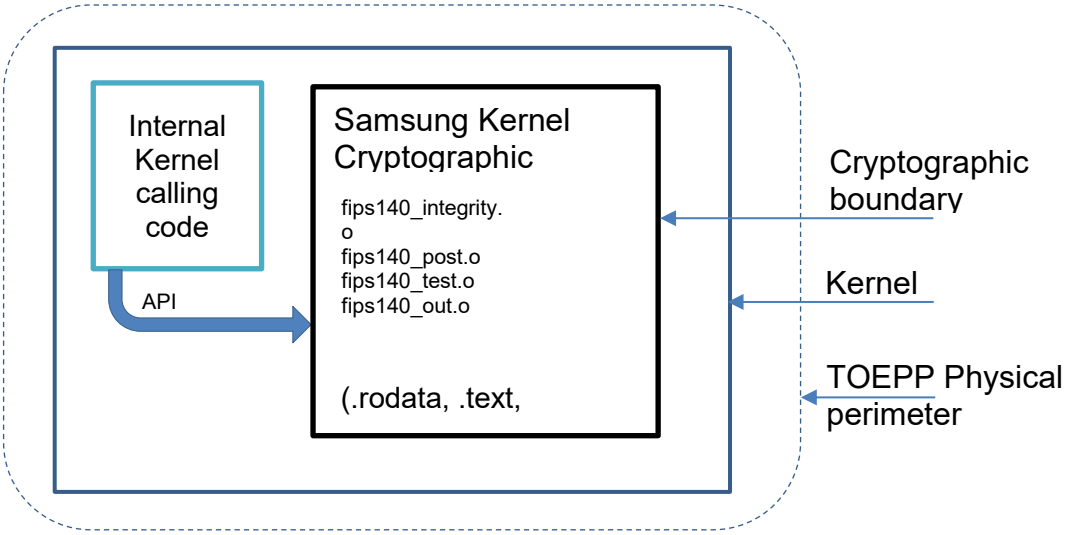

Figure 1: Block Diagram

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
boot.img	2.3		HMAC-SHA2-256

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)
Linux Kernel 5.15	Samsung Galaxy S23	Snapdragon 8 Gen 2	Yes		2.3
Linux Kernel 5.15	Samsung Galaxy S23	Snapdragon 8 Gen 2	No		2.3
Linux Kernel 5.15	Samsung Tab Active5	Samsung Electronics Exynos 1380	Yes		2.3

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Linux Kernel 5.15	Samsung Tab Active5	Samsung Electronics Exynos 1380	No		2.3
Linux Kernel 5.15	Samsung Galaxy Tab S9 FE	Samsung Electronics Exynos 1380	Yes		2.3
Linux Kernel 5.15	Samsung Galaxy Tab S9 FE	Samsung Electronics Exynos 1380	No		2.3

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 Galaxy 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

N/A for this module

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.API function skc_is_approved_service() returns 1.	Approved	Equivalent to the indicator of the requested service as defined in section 4.3
Non-Approved Mode	Automatically entered whenever a non-approved service is requested. API function skc_is_approved_service() returns 0.	Non-Approved	Equivalent to the indicator of the requested service as defined in section 4.4

Table 5: Modes List and Description

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. Table 16 lists the approved services. The non-

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

approved mode is entered when a non-approved algorithm/non-approved service is invoked. Table 17 lists non-approved services. The Approved mode of operation can only be transitioned into the non-Approved mode by calling one of the non-Approved services listed in Table 17.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3242	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A3242	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
HMAC-SHA-1	A3242	MAC - MAC: 160 Key Length - Key Length: 256, 384, 768, 1024	FIPS 198-1
HMAC-SHA2-224	A3242	MAC - MAC: 224 Key Length - Key Length: 256, 384, 768, 1024	FIPS 198-1
HMAC-SHA2-256	A3242	MAC - MAC: 256 Key Length - Key Length: 256, 384, 768, 1024	FIPS 198-1
HMAC-SHA2-384	A3242	MAC - MAC: 384 Key Length - Key Length: 256, 384, 1072, 1280	FIPS 198-1
HMAC-SHA2-512	A3242	MAC - MAC: 512 Key Length - Key Length: 256, 384, 1072, 1280	FIPS 198-1
SHA-1	A3242	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A3242	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3242	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A3242	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A3242	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
ESSIV-CBC-AES	Disk encryption/decryption with using AES (CBC) and SHA2-256
CMAC	Message authentication.
AES-CTR	Symmetric Encryption and Decryption.
AES-XTS	Disk encryption/decryption

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
AES encryption	BC-UnAuth	AES encryption		AES-CBC: (A3242) AES-ECB: (A3242)
AES decryption	BC-UnAuth	AES decryption		AES-CBC: (A3242) AES-ECB: (A3242)
HMAC generation	MAC	HMAC generation	Truncation:Not supported	HMAC-SHA-1: (A3242) HMAC-SHA2-224: (A3242) HMAC-SHA2-384: (A3242) HMAC-SHA2-512: (A3242) HMAC-SHA2-256: (A3242)
Hash generation	SHA	Hash generation		SHA-1: (A3242) SHA2-224: (A3242) SHA2-256: (A3242) SHA2-384: (A3242) SHA2-512: (A3242)

Table 8: Security Function Implementations

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

2.7 Algorithm Specific Information

- SHA-1

Per SP800-131Ar2, the use of SHA-1 is disallowed for digital signature generation, but is permitted for digital signature verification (legacy use) and all non-digital signature applications. This implementation will be non-Approved for all uses starting January 1, 2031. User should move to SHA2, which is available in this module.

2.8 RBG and Entropy

N/A for this module.

N/A for this module.

2.9 Key Generation

2.10 Key Establishment

2.11 Industry Protocols

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters.
N/A	Data Output	API output parameters.
N/A	Control Input	API function calls, API control input parameters
N/A	Control Output	N/A
N/A	Status Output	API return code, log messages.

Table 9: 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.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

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 10: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show status	Provide Module's current status (status message)	N/A	Command used to show Module's Status	Module's operational status	None	Crypto Officer
Show version	Provide Module's name and version information	N/A	Command to show Module's ID and version	Module's ID and versioning information	None	Crypto Officer
Perform Self-Tests	Perform Self-Tests (Pre-operational self-tests and Conditional Self-Tests)	N/A	Command to trigger self-tests	Status of the self-tests results	None	Crypto Officer
Perform Zeroization	Perform Zeroization	N/A	Command to zeroize	Status of the SSPs	None	Crypto Officer - AES

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			the module	zeroization		Key: Z - HMAC Key: Z
Module initialization	Initialize the module	N/A	Command to initialize the module	Initialization status	None	Crypto Officer
Symmetric encryption	Encrypt a plaintext	API function get_service_status() returns 1	Key, cipher text, initialization vector	Cipher text	AES encryption	Crypto Officer - AES Key: W,E
Symmetric decryption	Decrypt a cipher text	API function get_service_status() returns 1	Key, cipher text, initialization vector	Plain text	AES decryption	Crypto Officer - AES Key: W,E
Message digest generation	Generate message digest	API function get_service_status() returns 1	Message	Message digest	Hash generation	Crypto Officer
MAC generation	Generate message authentication code	API function get_service_status() returns 1	Message, key	MAC	HMAC generation	Crypto Officer - HMAC Key: W,E

Table 11: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
MAC generation	Generate message authentication code. API function get_service_status() returns 0	CMAC	Crypto Officer
Symmetric encryption/decryption	Non-approved encryption/decryption algorithm. API function get_service_status() returns 0	ESSIV-CBC-AES AES-CTR	Crypto Officer
Symmetric encryption/decryption with block stealing.	Non-approved symmetric, block stealing, encryption/decryption algorithm. API function get_service_status() returns 0	ESSIV-CBC-AES AES-XTS	Crypto Officer

Table 12: Non-Approved Services

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

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. #A3242) algorithm. The software integrity test key (non-SSP) was preloaded to the module's binary in 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 a HMAC-SHA2-256 digest which is compared to a value computed at build time. If at load time the MAC does not match the stored, known MAC value, the module would enter to an Error state with all crypto functionality inhibited.

5.2 Initiate on Demand

The integrity test is performed as part of the pre-operational self-tests. It is automatically executed at power-on. The operator can initiate the integrity test on demand by power cycling the host platform.

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.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

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
Tested Platform's RAM (Volatile memory)	Temporary storage within TOEPP for SSPs used by the module as part of service execution	Dynamic

Table 13: Storage Areas

The module does not provide persistent storage for keys or SSPs. The module uses pointers to plaintext keys/SSPs that are passed in by the calling application. The module does not store any SSP beyond the lifetime of an API call. Allocated memory in RAM for SSP is managed by the module.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
SSPs API input	Calling application (TOEPP)	Module	Plaintext	Manual	Electronic	

Table 14: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroization API call	Zeroization command calling context destructor to zeroize all SSPs stored in its context struct	SSPs are actively overwritten with zeroes and thus not recoverable	Using crypto_free_<transformation type>()
Power down	Power down the tested platform to zeroize all SSPs	SSPs stored in the tested platform's volatile memory will	Power down the tested platform

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Zeroization Method	Description	Rationale	Operator Initiation
		be zeroized after power is lost	

Table 15: SSP Zeroization Methods

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			AES encryption AES decryption
HMAC Key	Keyed Hash	at least 160 bits - at least 112 bits	MAC key - CSP			HMAC generation

Table 16: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	SSPs API input	Tested Platform's RAM (Volatile memory):Plaintext	For the lifetime of the API call	Zeroization API call Power down	
HMAC Key	SSPs API input	Tested Platform's RAM (Volatile memory):Plaintext	For the lifetime of the API call	Zeroization API call Power down	

Table 17: SSP Table 2

The cryptographic module is passed a pointer to the cryptographic keys as API parameters, associated by memory location. The application calling the cryptographic module passes keys in plaintext within the physical perimeter. The module does not perform storage of keys. All SSPs can be zeroized by power cycling the host.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3242)	HMAC-SHA2-256	KAT	SW/FW Integrity	When the test passes, do_integrity_check() returns 0,	HMAC-SHA2-256

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
				otherwise it returns -1 and device gets into error state.	

Table 18: 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). 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 KAT (A3242)	Key lengths: 128, 192, 256 bits	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	AES-ECB encryption	During the module start-up or using fips140_kat()
AES-ECB Decrypt KAT (A3242)	Key lengths: 128, 192, 256 bits	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	AES-ECB decryption	During the module start-up or using fips140_kat()
SHA-1 KAT (A5143)	N/A	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	Hash generation	During module start-up or on-demand using fips140_kat()
SHA2-224 KAT (A5143)	N/A	KAT	CAST	When the test passes, fips140_kat() returns 0,	Hash generation	During module start-up or on-demand using fips140_kat()

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				otherwise it returns -1 and device gets into error state.		
SHA2-256 KAT (A5143)	N/A	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	Hash generation	During module start-up or on-demand using fips140_kat()
SHA2-384 KAT (A5143)	N/A	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	Hash generation	During module start-up or on-demand using fips140_kat()
SHA2-512 KAT (A5143)	N/A	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	Hash generation	During module start-up or on-demand using fips140_kat()
HMAC-SHA-1 KAT (A3242)	SHA-1	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	HMAC generation	During module start-up or on-demand using fips140_kat()
HMAC-SHA2-224 KAT (A3242)	SHA2-224	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	HMAC generation	During module start-up or on-demand using fips140_kat()
HMAC-SHA2-256 KAT (A3242)	SHA2-256	KAT	CAST	When the test passes, fips140_kat() returns 0,	HMAC generation	During module start-up or on-demand using fips140_kat()

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				otherwise it returns -1 and device gets into error state.		
HMAC-SHA2-384 KAT (A3242)	SHA2-384	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	HMAC generation	During module start-up or on-demand using fips140_kat()
HMAC-SHA2-512 KAT (A3242)	SHA2-512	KAT	CAST	When the test passes, fips140_kat() returns 0, otherwise it returns -1 and device gets into error state.	HMAC generation	During module start-up or on-demand using fips140_kat()

Table 19: Conditional Self-Tests

The module performs on-demand self-tests initiated by the operator, by powering off and powering the module back on. The full suite of self-tests is then executed. The same procedure may be employed by the operator to perform periodic self-tests.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3242)	KAT	SW/FW Integrity	On demand	Reload the module or use fips140_kat() API call

Table 20: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB Encrypt KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
AES-ECB Decrypt KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA-1 KAT (A5143)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
SHA2-224 KAT (A5143)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
SHA2-256 KAT (A5143)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
SHA2-384 KAT (A5143)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
SHA2-512 KAT (A5143)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
HMAC-SHA-1 KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
HMAC-SHA2-224 KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
HMAC-SHA2-256 KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
HMAC-SHA2-384 KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call
HMAC-SHA2-512 KAT (A3242)	KAT	CAST	On demand	Reload the module or use fips140_kat() API call

Table 21: Conditional Periodic Information

The module performs on-demand self-tests initiated by the operator (via calling fips140_post() service), by power-cycling, or rebooting the tested platform. The pre-operational software integrity test and the full suite of self-tests listed in Table 12 are executed. The same procedure may be employed by the operator to perform periodic self-tests. If any of the tests fail, the module will enter error state.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

10.4 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	Reload the module	The OE`s OS log contains message 'FIPS : POST - one or more selftests failed'.

Table 22: Error States

The error state represents an unrecoverable error. The module will automatically reload the image. In the Error State, no cryptographic services are provided, and data output is prohibited.

The module has a variable (skc_fips_enabled) indicating the status of the Self-test. It contains 0 if the Self-test was failed and it contains 1 if the Self-test was successful.

The kernel logs contains message:

FIPS : POST - integrity test failed

in case of integrity test is failed,

FIPS : <alg name>, test failed, err=<test number>

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

This cryptographic module is built-in along with the Linux Kernel. The module is initialized during the Kernel boot-up before any cryptographic functionality is available. The Kernel is responsible for the initialization and loading processes of the module. The module is designed with module init entry point which ensures that the pre-operational self-tests and CASTs are initiated automatically when the module is loaded.

11.2 Administrator Guidance

N/A

11.3 Non-Administrator Guidance

N/A

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice

12 Mitigation of Other Attacks

N/A for this module.

©2026 Samsung Electronics Co., Ltd.
This document can be reproduced and distributed only whole and intact, including this copyright notice