Cloud Linux Software, Inc. d/b/a TuxCare

TuxCare Kernel Crypto API

FIPS 140-3 Non-Proprietary Security Policy

Prepared by:

atsec information security corporation
4516 Seton Center Pkwy, Suite 250
Austin, TX 78759
www.atsec.com

Document version: 1.0
Last update: 2026-06-08

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 version kernel 5.14.0-570.62.1.el9_6.tuxcare.5; libkcapi 1.4.0-2.el9_6.tuxcare.1 of the TuxCare Kernel Crypto API 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.

This Non-Proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

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

In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The TuxCare Kernel Crypto API module (hereafter referred to as "the module") provides a C language application program interface (API) for use by other (kernel space and user space) processes that require cryptographic functionality. The module operates on a general-purpose computer as part of the Linux kernel. Its cryptographic functionality can be
accessed using the Linux Kernel Crypto API.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the kernel binary and the kernel crypto object files, the libkcapi library, and the sha512hmac binary, which is used to verify the integrity of the software components. In addition, the cryptographic boundary contains the .hmac files which store the expected integrity values for each of the software components.

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP of the module is defined as the general-purpose computer on which the module is installed. The PAA provided by the processor is located within the module's physical perimeter and outside of the module's cryptographic boundary. The TOEPP is indicated by the large thin border in Figure 1.

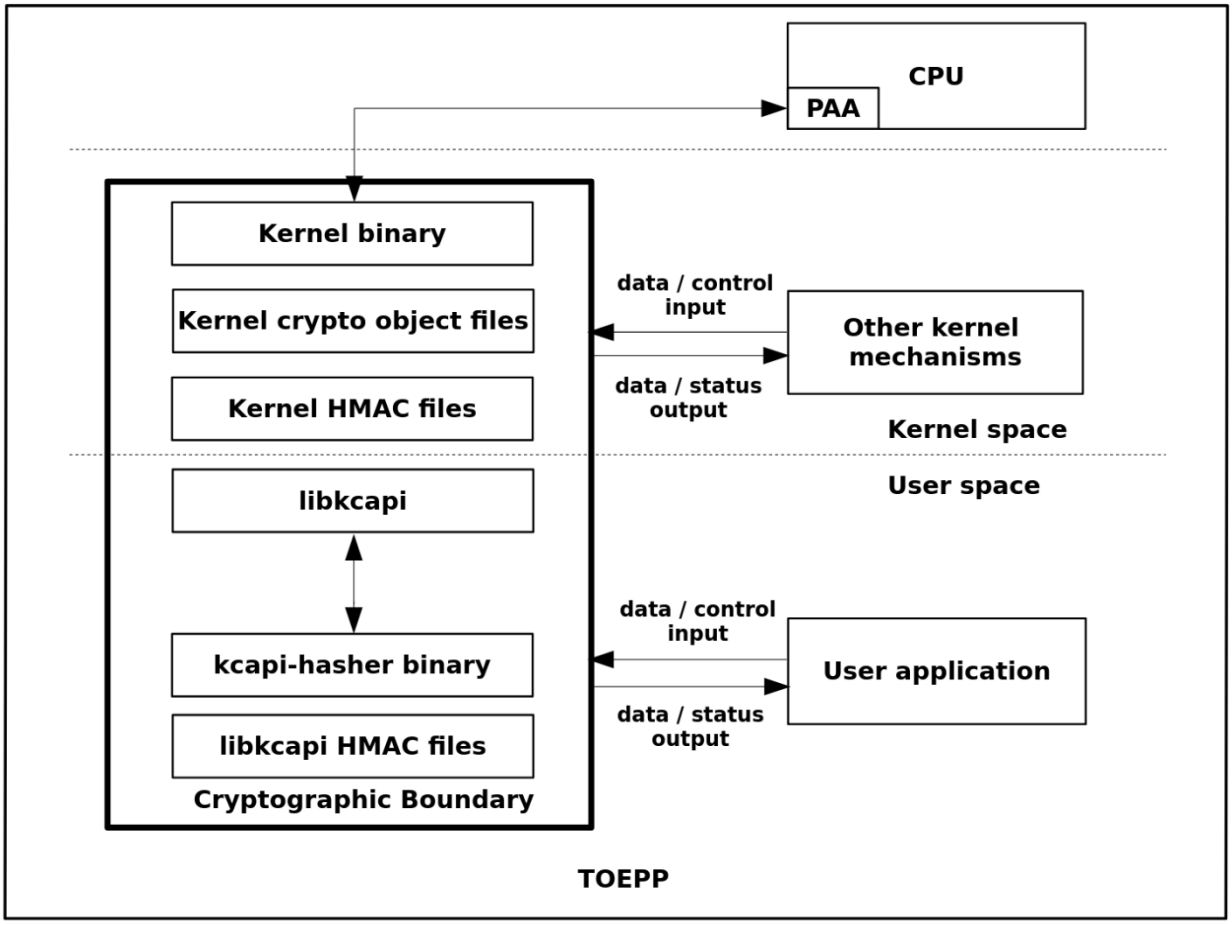

Figure 1: Block Diagram

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/vmlinuz-5.14.0-570.62.1.el9_6.tuxcare.5.x86_64; *.ko files in /usr/lib/modules/5.14.0-570.62.1.el9_6.tuxcare.5.x86_64/kernel/crypto/; *ko files in /usr/lib/modules/5.14.0-570.62.1.el9_6.tuxcare.5.x86_64/kernel/arch/x86/crypto/; /usr/lib64/libkcapi.so.1.4.0, /usr/bin/sha512hmac	Kernel: 5.14.0-570.62.1.el9_6.tuxcare.5; libkcapi: 1.4.0-2.el9_6.tuxcare.1	N/A	HMAC-SHA2-512 (vmlinuz, libkcapi.so.1.4.0, sha512hmac); RSA signature

Package or File Name	Software/ Firmware Version	Features	Integrity Test
			verification (*.ko files)

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)
AlmaLinux OS 9.6	GIGABYTE E163-S30-AAG1	Intel® Xeon® Gold 5512U	Yes	N/A	5.14.0-570.62.1.el9_6.tuxcare.51.4.0-2.el9_6.tuxcare.1
AlmaLinux OS 9.6	GIGABYTE E163-S30-AAG1	Intel® Xeon® Gold 5512U	No	N/A	5.14.0-570.62.1.el9_6.tuxcare.51.4.0-2.el9_6.tuxcare.1

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

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

Operating System	Hardware Platform
Rocky Linux 9.6	GIGABYTE E163-S30-AAG1 on Intel® Xeon® Gold 5512U

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

There are no components within the cryptographic boundary excluded from the FIPS 140-3 requirements.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved	Automatically entered whenever	Approved	Mapped to approved service indicator in Section 4.3 for all approved algorithms except GCM: respective approved

Mode Name	Description	Type	Status Indicator
	an approved service is requested.		service function returns indicator 0. For GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set
Non-Approved	Automatically entered whenever a non-approved service is requested.	Non-Approved	No service indicator required for non-approved services per IG 2.4.C

Table 5: Modes List and Description

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode. No operator intervention is required to reach this point.

Mode Change Instructions and Status:

The module automatically switches between the approved and non-approved modes depending on the services requested by the operator. The status indicator of the mode of operation is equivalent to the indicator of the service that was requested.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7051, A7056, A7059	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A7051, A7056, A7059	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A7051, A7059	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A7051, A7059	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A7051, A7059	Direction - Generation Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A7051, A7056, A7059	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A7051, A7054, A7056, A7058, A7059, A7061	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A7053, A7057, A7060	Direction - Encrypt IV Generation - Internal	SP 800-38D

Algorithm	CAVP Cert	Properties	Reference
		IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	
AES-GMAC	A7051, A7056, A7059	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-OFB	A7051, A7059	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A7051, A7056, A7059	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA SigVer (FIPS186-4)	A7055	Component - No Curve - P-256, P-384 Hash Algorithm - SHA-1	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A7055	Curve - P-256, P-384 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
Hash DRBG	A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061, A7062, A7063, A7064	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061, A7062, A7063, A7064	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A7051, A7062, A7063, A7064	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A7051, A7062, A7063, A7064	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A7051, A7062, A7063, A7064	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A7051, A7062, A7063, A7064	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A7051, A7062, A7063, A7064	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A7052	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A7052	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A7052	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A7052	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-5)	A7051, A7062, A7063, A7064	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
SHA2-224	A7051, A7062, A7063, A7064	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-256	A7051, A7062, A7063, A7064	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-384	A7051, A7062, A7063, A7064	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-512	A7051, A7062, A7063, A7064	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA3-224	A7052	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-256	A7052	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-384	A7052	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-512	A7052	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

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
AES GCM with external IV	Encryption with external IV (not compliant to FIPS 140-3 IG C.H)
KBKDF (libkcapi)	Key derivation with implementation not tested by CAVP
HKDF (libkcapi)	Key derivation with implementation not tested by CAVP
PBKDF2 (libkcapi)	Password-based Key derivation with implementation not tested by CAVP
RSA	Encryption primitive; Decryption primitive (not compliant to SP 800-56Br2)
RSA with PKCS#1 v1.5 padding	Signature generation(pre-hashed message) primitive; Signature verification(pre-hashed message) primitive
ECDSA	Signature generation(pre-hashed message) primitive; Signature verification(pre-hashed message) primitive

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric Encryption with AES	BC-UnAuth	Symmetric Encryption using AES		AES-CBC: (A7051, A7056, A7059) AES-CBC-CS3: (A7051, A7056, A7059) AES-CFB128: (A7051, A7059) AES-CTR: (A7051, A7056, A7059) AES-ECB: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061) AES-OFB: (A7051, A7059) AES-XTS Testing Revision 2.0: (A7051, A7056, A7059)
Authenticated Symmetric Encryption with AES	BC-Auth	Encrypt and authenticate a plaintext with AES		AES-CCM: (A7051, A7059) AES-GCM: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061)

Name	Type	Description	Properties	Algorithms
Symmetric Decryption with AES	BC-UnAuth	Symmetric Decryption using AES		AES-CBC: (A7051, A7056, A7059) AES-CBC-CS3: (A7051, A7056, A7059) AES-CFB128: (A7051, A7059) AES-CTR: (A7051, A7056, A7059) AES-ECB: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061) AES-OFB: (A7051, A7059) AES-XTS Testing Revision 2.0: (A7051, A7056, A7059)
Authenticated Symmetric Decryption with AES	BC-Auth	Decrypt and authenticate a ciphertext with AES		AES-CCM: (A7051, A7059) AES-GCM: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061)
Message Digest with SHA-3	SHA	Compute a message digest		SHA3-224: (A7052) SHA3-256: (A7052) SHA3-384: (A7052) SHA3-512: (A7052)
Message Digest with SHA-2	SHA	Compute a message digest		SHA2-224: (A7051, A7062, A7063, A7064) SHA2-256: (A7051, A7062, A7063, A7064) SHA2-384: (A7051, A7062, A7063, A7064) SHA2-512: (A7051, A7062, A7063, A7064)
Random Number Generation with CTR_DRBG	DRBG	Generate random bitstrings		Counter DRBG: (A7051, A7053, A7054, A7056,

Name	Type	Description	Properties	Algorithms
				A7057, A7058, A7059, A7060, A7061)
Random Number Generation with HMAC_DRBG	DRBG	Generate random bitstrings		HMAC DRBG: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061, A7062, A7063, A7064)
Random Number Generation with Hash_DRBG	DRBG	Generate random bitstrings		Hash DRBG: (A7051, A7053, A7054, A7056, A7057, A7058, A7059, A7060, A7061, A7062, A7063, A7064)
Message Authentication Code (MAC) with HMAC	MAC	Compute a MAC tag		HMAC-SHA-1: (A7051, A7062, A7063, A7064) HMAC-SHA2-224: (A7051, A7062, A7063, A7064) HMAC-SHA2-256: (A7051, A7062, A7063, A7064) HMAC-SHA2-384: (A7051, A7062, A7063, A7064) HMAC-SHA2-512: (A7051, A7062, A7063, A7064) HMAC-SHA3-224: (A7052) HMAC-SHA3-256: (A7052) HMAC-SHA3-384: (A7052) HMAC-SHA3-512: (A7052)
Message Authentication Code (MAC) with AES	MAC	Compute a MAC tag		AES-CMAC: (A7051, A7059) AES-GMAC: (A7051, A7056, A7059)

Name	Type	Description	Properties	Algorithms
Signature verification with ECDSA	DigSig-SigVer	Internal function for digital signature verification using ECDSA with SHA2-224, SHA2-256, SHA2-384, SHA2-512		ECDSA SigVer (FIPS186-5): (A7055)
Signature verification with ECDSA (legacy use)	DigSig-SigVer	Internal function for digital signature verification using ECDSA with SHA-1	Publications:FIPS 140-3 IG C.M legacy algorithms	ECDSA SigVer (FIPS186-4): (A7055)
Signature Verification with RSA	DigSig-SigVer	Internal function for RSA signature verification for integrity test for kernel crypto object files		RSA SigVer (FIPS186-5): (A7051, A7062, A7063, A7064)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

The Crypto Officer shall consider the following requirements and restrictions when using the module.

For IPsec, the module offers the AES GCM implementation and uses the context of Scenario 1 of FIPS 140-3 IG C.H. The mechanism for IV generation is compliant with RFC 4106. IVs generated using this mechanism may only be used in the context of AES GCM encryption within the IPsec protocol.

The module does not implement IPsec. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. This application must use RFC 7296 compliant IKEv2 to establish the shared secret SKEYSEED from which the AES GCM encryption keys are derived.

The design of the IPsec protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES GCM key encryption or decryption under this scenario shall be established.

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. This service can be requested by invoking the crypto_aead_encrypt API function with an AES GCM handle. When this is the case, the API will not set an approved service indicator.

2.7.2 AES XTS

The length of a single data unit encrypted or decrypted with AES XTS shall not exceed 2^20 AES blocks, that is 16MB, of data per XTS instance. An XTS instance is defined in Section 4 of SP 800-38E.

The XTS mode shall only be used for the cryptographic protection of data on storage devices. It shall not be used for other purposes, such as the encryption of data in transit.

To meet the requirement stated in IG C.I, the module implements a check to ensure that the two AES keys used in AES XTS mode are not identical. As the module does not implement symmetric key generation, this check is performed when the keys are input by the operator. Key_1 and Key_2 shall be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133r2, Section 6.3.

2.7.3 RSA

The module supports RSA modulus lengths of 2048, 3072, and 4096 bits for signature verification. The RSA signature verification implementation has been tested for all implemented RSA modulus lengths.

2.7.4 Authenticated Encryption / Decryption

The module does not establish SSP's using an approved key transport scheme (KTS).
However, it does offer approved authenticated algorithms that can be used by an external operator /application as part of an approved KTS.

2.7.5 Legacy Use

ECDSA Digital signature verification using SHA-1 is allowed for legacy use only. Algorithms designated as "Legacy" can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

2.8 RBG and Entropy

Cert Number	Vendor Name
E75	Cloud Linux Software, Inc. d/b/a TuxCare

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Cloudlinux Inc., TuxCare division Kernel CPU Time	Non-Physical	AlmaLinux OS 9.6 on GIGABYTE E163-S30-	256 bits	256 bits	SHA3-256 (Cert. A7065)

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Jitter RNG Entropy Source		AAG1 on Intel® Xeon® Gold 5512U			

Table 10: Entropy Sources

The module implements three different Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90Ar1: CTR_DRBG, Hash_DRBG, and HMAC_DRBG. Each of these DRBG implementations can be instantiated by the operator of the module. When instantiated, these DRBGs can be used to generate random numbers for external usage.

The DRBG is initially seeded and reseed with full entropy outputs obtained from the entropy source. The module complies with the Public Use Document for ESV certificate E75 by reading entropy data from the jent_kcapi_random() function, which corresponds to the GetEntropy() conceptual interface. This function outputs 256 bits of full entropy.

The operational environment on the ESV certificate is identical to the operating system described in this document, and the entropy source is implemented inside the cryptographic boundary. Thus, the module is compliant with scenario 1 of IG 9.3.A. There are no maintenance requirements for the entropy source.

2.9 Key Generation

The module does not implement any SSP generation methods.

2.10 Key Establishment

The module does not implement any SSP establishment methods.

2.11 Industry Protocols

AES GCM with internal IV generation in approved mode is compliant with RFC 4106 and shall only be used in conjunction with the IPsec protocol. No parts of this protocol, other than the AES GCM implementation, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API data input parameters, AF_ALG type sockets
N/A	Data Output	API output parameters, AF_ALG type sockets
N/A	Control Input	API function calls, API control input parameters, AF_ALG type sockets, kernel command line
N/A	Status Output	API return values, AF_ALG type sockets, kernel logs

Table 11: Ports and Interfaces

The logical interfaces are the APIs through which the applications request services and AF_ALG type socket that allows the applications running in the user space to request cryptographic services from the module. These logical interfaces are logically separated from each other by the API design.

The module does not support a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement authentication.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	None

Table 12: Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module. No support is provided for multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric Encryption	Symmetric Encryption using AES	crypto_skcipher_setkey return 0	Plaintext, AES key	Ciphertext	Symmetric Encryption with AES	Crypto Officer - AES key: W,E
Symmetric Decryption	Symmetric Decryption using AES	crypto_skcipher_setkey return 0	Ciphertext, AES key	Plaintext	Symmetric Decryption with AES	Crypto Officer - AES key: W,E
Authenticated Symmetric Encryption	Encrypt and authenticate a plaintext	For all except AES GCM: crypto_aead_setkey returns 0; For AES GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set	Plaintext, AES key, IV	Ciphertext, MAC tag	Authenticated Symmetric Encryption with AES	Crypto Officer - AES key: W,E - GCM IV: G,R,E
Authenticated Symmetric Decryption	Decrypt and authenticate a ciphertext	For all except AES GCM: crypto_aead_setkey returns 0; For AES GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set	Ciphertext, AES key, MAC tag, IV	Plaintext	Authenticated Symmetric Decryption with AES	Crypto Officer - AES key: W,E - GCM IV: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Authentication Code (MAC)	Compute a MAC tag	crypto_shash_init returns 0	AES: AES key, message; HMAC: HMAC key, message	MAC tag	Message Authentication Code (MAC) with AES Message Authentication Code (MAC) with HMAC	Crypto Officer - AES key: W,E - HMAC key: W,E
Random Number Generation	Generate random bitstrings	crypto_rng_get_bytes returns 0	Output length	Random bytes	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG	Crypto Officer - Entropy input: G,E,Z - CTR_DRBG Seed (IG D.L): G,E - Hash_DRBG Seed (IG D.L): G,E - HMAC_DRBG Seed (IG D.L): G,E - CTR_DRBG Internal State (V, Key) (IG D.L): G,W,E - Hash_DRBG Internal State (V, C) (IG D.L): G,W,E - HMAC_DRBG Internal State (V, Key) (IG D.L): G,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Random number generation using entropy source	Get entropy from a dedicated instance of the Jitter	crypto_rng_get_bytes returns 0	Output length	Random bytes	None	Crypto Officer
Message digest	Compute a message digest	crypto_shash_init returns 0	Message	Digest value	Message Digest with SHA-2 Message Digest with SHA-3	Crypto Officer
Compression	Compress data (deflate, lzo, zlib-deflate)	None	Data	Compressed data	None	Crypto Officer
Generic system call	Use the kernel to perform various non-cryptographic operations	None	Identifier, various arguments	Various return values	None	Crypto Officer
Error detection code	Compute an EDC (crc32, crct10dif)	None	Message	EDC	None	Crypto Officer
Show status	Return the module status	None	N/A	Module status	None	Unauthenticated
Show module name and version	Return the module name and version information	None	N/A	Module name and version	None	Unauthenticated
Self-test	Perform the CASTs and	None	N/A	Pass/fail result of self-tests	Symmetric Encryption with AES Authentica	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	integrity tests				ted Symmetric Encryption with AES Symmetric Decryption with AES Authenticated Symmetric Decryption with AES Message Digest with SHA-3 Message Digest with SHA-2 Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG Message Authentication Code (MAC) with HMAC Message Authentication Code	

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					(MAC) with AES Signature verification with ECDSA Signature verification with ECDSA (legacy use) Signature Verification with RSA	
Zeroization	Zeroize SSPs	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - Entropy input: Z - CTR_DRBG Seed (IG D.L): Z - Hash_DRBG Seed (IG D.L): Z - HMAC_DRBG Seed (IG D.L): Z - CTR_DRBG Internal State (V, Key) (IG D.L): Z - Hash_DRBG Internal State (V, C) (IG D.L): Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- HMAC_DRBG Internal State (V, Key) (IG D.L): Z - GCM IV: Z

Table 13: Approved Services

The above table lists the approved services. The following convention is used to specify access rights to SSPs:

- **Generate (G)**: The module generates or derives the SSP.
- **Read (R)**: The SSP is read from the module (e.g., the SSP is output).
- **Write (W)**: The SSP is updated, imported, or written to the module.
- **Execute (E)**: The module uses the SSP in performing a cryptographic operation.
- **Zeroize (Z)**: The module zeroizes the SSP.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM with external IV	Encryption	AES GCM with external IV	CO
KBKDF (libkcapi)	Key derivation	KBKDF (libkcapi)	CO
HKDF (libkcapi)	Key derivation	HKDF (libkcapi)	CO
PBKDF2 (libkcapi)	Password-based key derivation	PBKDF2 (libkcapi)	CO
RSA	Encryption primitive; Decryption primitive	RSA	CO
RSA with PKCS#1 v1.5 padding	Signature generation (pre-hashed message); Signature verification (pre-hashed message)	RSA with PKCS#1 v1.5 padding	CO
ECDSA	Signature generation (pre-hashed message); Signature verification (pre-hashed message)	ECDSA	CO

Table 14: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The static kernel binary, libkcapi, and sha512hmac software components are integrity tested using an HMAC-SHA2-512 calculation performed by the sha512hmac utility (which utilizes the module's HMAC and SHA-512 implementations). The kernel crypto object files listed in Table 2 are loaded on start-up by the module and verified using RSA signature verification with PKCS#1 v1.5 padding, SHA-256, and a 4096-bit key.

5.2 Initiate on Demand

Integrity tests are performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity tests can be invoked on demand by unloading and subsequently re-initializing the module, which will perform (among others) the software integrity tests.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The approved cryptographic algorithms of the module are part of the Linux kernel, which operates in Linux kernel space. This ensures that any SSPs contained within the module are protected by the process isolation and memory separation mechanisms provided by the Linux kernel, and only the module has control over these SSPs. The user space libkcapi and sha512hmac components, though not processing any SSPs, are similarly protected by the operating environment.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11.1.

Instrumentation tools like the ptrace system call, gdb and strace, as well as other tracing mechanisms offered by the Linux environment such as ftrace or systemtap, shall not be used in the operational environment. The use of any of these tools implies that the cryptographic module is running in a non-validated operational environment.

7 Physical Security

The module is comprised of software only and therefore this section is not applicable.

8 Non-Invasive Security

This module does not implement any non-invasive security mechanism and therefore this section is not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution.	Dynamic

Table 15: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form. SSPs are provided to the module by the calling process and are destroyed when released by the appropriate zeroization function calls.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input parameters; AF_ALG_type sockets (input)	Operator calling application (TOEPP)	Cryptographic module	Plaintext	Manual	Electronic	
API output parameters; AF_ALG_type sockets (output)	Cryptographic module	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	

Table 16: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the cipher handle.	Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization procedure succeeded.	By calling the appropriate cipher related zeroization API functions: AES key: crypto_free_skcipher and crypto_free_aead; HMAC key: crypto_free_shash and crypto_free_ahash; DRBG internal state: crypto_free_rng; EC public key, RSA public key: public_key_free
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A

Zeroization Method	Description	Rationale	Operator Initiation
Remove power from the module	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed. Module power off indicates that the zeroization procedure succeeded. The successful removal of power implicitly indicates that the zeroization is complete.	By removing power

Table 17: SSP Zeroization Methods

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key used for Encryption; Decryption; Authenticated encryption; Authenticated decryption; Message authentication	XTS: 128, 256 bits; Other modes: 128, 192, 256 bits - XTS: 128, 256 bits; Other modes: 128, 192, 256 bits	Symmetric key - CSP			Symmetric Encryption with AES Authenticated Symmetric Encryption with AES Symmetric Decryption with AES Authenticated Symmetric Decryption with AES Message Authentication Code (MAC) with AES
HMAC key	HMAC key used for Message Authentication	112-524288 bits - 112-256 bits	Authentication key - CSP			Message Authentication Code (MAC) with HMAC
Entropy input	Entropy input used for Random	128-384 bits -	Entropy Input - CSP			Random Number Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	number generation	128-384 bits				with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
CTR_DRBG Seed (IG D.L)	DRBG seed derived from Entropy Input	256, 320, 384 bits - 128, 192, 256 bits	Seed - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
Hash_DRBG Seed (IG D.L)	DRBG seed derived from Entropy Input	440, 888 bits - 128, 256 bits	Seed - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC_DRBG Seed (IG D.L)	DRBG seed derived from Entropy Input	440, 888 bits - 128, 256 bits	Seed - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
CTR_DRBG Internal State (V, Key) (IG D.L)	Internal state of Counter DRBG instance	256, 320, 384 bits - 128, 192, 256 bits	Internal state - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
Hash_DRBG Internal State (V, C) (IG D.L)	Internal state of Hash DRBG instance	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
				Number Generation with Hash_DRBG		Generation with Hash_DRBG
HMAC_DRBG Internal State (V, Key) (IG D.L)	Internal state of HMAC DRBG instance	320, 512, 1024 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG		Random Number Generation with CTR_DRBG Random Number Generation with HMAC_DRBG Random Number Generation with Hash_DRBG
GCM IV	IV used for Authenticated encryption; Authenticated decryption	96 bit - N/A	Initialization vector - PSP	Authenticated Symmetric Encryption with AES		Authenticated Symmetric Encryption with AES Authenticated Symmetric Decryption with AES

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	
HMAC key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Entropy input		RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	CTR_DRBG Seed (IG D.L):Derives Hash_DRBG Seed (IG D.L):Derives HMAC_DRBG Seed (IG D.L):Derives
CTR_DRBG Seed (IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic Remove power from the module	Entropy Input (IG D.L):Derived From CTR_DRBG Internal State (V, Key) (IG D.L):Derives
Hash_DRBG Seed (IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic Remove power from the module	Entropy Input (IG D.L):Derived From Hash_DRBG Internal State (V, C) (IG D.L):Derives
HMAC_DRBG Seed (IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic Remove power from the module	Entropy Input (IG D.L):Derived From HMAC_DRBG Internal State (V, Key) (IG D.L):Derives
CTR_DRBG Internal State (V, Key) (IG D.L)		RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	CTR_DRBG Seed (IG D.L):Derived From
Hash_DRBG Internal State (V, C) (IG D.L)		RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	Hash_DRBG Seed (IG D.L):Derived From
HMAC_DRBG Internal State (V, Key) (IG D.L)		RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	HMAC_DRBG Seed (IG D.L):Derived From
GCM IV	API input parameters;	RAM:Plaintext	From service invocation to	Free cipher handle	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	AF_ALG_type sockets (input) API output parameters; AF_ALG_type sockets (output)		service completion	Automatic Remove power from the module	

Table 19: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2031.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-512 (A7064)	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for kernel binary, libkcapi, and sha512hmac software components
RSA SigVer (FIPS186-5) (A7051)	4096-bit key with SHA-256	Signature Verification	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for kernel crypto object files

Table 20: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is powered on, before the module transitions into the operational state. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-224 (A7051)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A7062)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A7063)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-224 (A7064)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A7051)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A7062)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A7063)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A7064)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A7051)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A7062)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A7063)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
SHA2-384 (A7064)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A7051)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A7062)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A7063)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A7064)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-224 (A7052)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-256 (A7052)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA3-384 (A7052)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-512 (A7052)	0-8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
AES-ECB - Encrypt (A7051)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7051)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Encrypt (A7053)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7053)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Encrypt (A7054)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7054)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
AES-ECB - Encrypt (A7056)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7056)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Encrypt (A7057)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7057)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Decrypt (A7058)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB - Encrypt (A7060)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7060)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-ECB - Encrypt (A7061)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-ECB - Decrypt (A7061)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CBC - Encrypt (A7051)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use	Encryption	Module initialization
AES-CBC - Decrypt (A7051)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use	Decryption	Module initialization
AES-CBC - Encrypt (A7056)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-CBC - Decrypt (A7056)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
AES-CBC - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-CBC - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CFB128 - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-CFB128 - Decrypt (A7051)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CFB128 - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CBC-CS3 - Encrypt (A7059)	Encrypt with 128 bit keys	KAT	CAST	Module becomes operational and services are available for use	Encryption	Module initialization
AES-CBC-CS3 - Decrypt (A7059)	Decrypt with 128 bit keys	KAT	CAST	Module becomes operational and services are available for use	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-OFB - Encrypt (A7059)	128 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-OFB - Decrypt (A7059)	128 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7051)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use	Encryption	Module initialization
AES-GCM - Decrypt (A7051)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use	Decryption	Module initialization
AES-GCM - Encrypt (A7053)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7053)	Decrypt with 128, 192, 256 bit keys; external 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7054)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7054)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
AES-GCM - Encrypt (A7056)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7056)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7057)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7057)	Decrypt with 128, 192, 256 bit keys; external 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7058)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7058)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use	Decryption	Module initialization
AES-GCM - Encrypt (A7060)	Encrypt with 128, 192, 256 bit keys; external 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7060)	Decrypt with 128, 192, 256 bit keys; external 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-GCM - Encrypt (A7061)	Encrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM - Decrypt (A7061)	Decrypt with 128, 192, 256 bit keys; Internal 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-XTS Testing Revision 2.0 - Encrypt (A7051)	Encrypt with 128, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-XTS Testing Revision 2.0 - Decrypt (A7051)	Decrypt with 128, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-XTS Testing Revision 2.0 -	Encrypt with 128, 256 bit keys	KAT	CAST	Module becomes operational	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Encrypt (A7059)				and services are available for use.		
AES-XTS Testing Revision 2.0 - Decrypt (A7059)	Decrypt with 128, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CMAC (A7059)	128 and 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
AES-CTR - Encrypt (A7051)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-CTR - Decrypt (A7051)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CTR - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-CTR - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
AES-CCM - Encrypt (A7059)	Encrypt with 128, 192, 256 bit keys; 128-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CCM - Decrypt (A7059)	Decrypt with 128, 192, 256 bit keys; 128-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Decryption	Module initialization
HMAC-SHA-1 (A7051)	32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A7062)	32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A7063)	32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A7064)	32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A7051)	SHA2-224, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A7062)	SHA2-224, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A7063)	SHA2-224, 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
HMAC-SHA2-224 (A7064)	SHA2-224, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A7051)	SHA2-256, 32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A7062)	SHA2-256, 32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A7063)	SHA2-256, 32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A7064)	SHA2-256, 32-64 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A7051)	SHA2-384, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A7062)	SHA2-384, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-384 (A7063)	SHA2-384, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A7064)	SHA2-384, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-512 (A7051)	SHA2-512, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA2-512 (A7062)	SHA2-512, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA2-512 (A7063)	SHA2-512, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA2-512 (A7064)	SHA2-512, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA3-224 (A7052)	SHA3-224, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA3-256 (A7052)	SHA3-256, 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use.		
HMAC-SHA3-384 (A7052)	SHA3-384, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA3-512 (A7052)	SHA3-512, 32-1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
Counter DRBG (A7051)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7053)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7054)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7056)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7057)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG (A7058)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7059)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7060)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Counter DRBG (A7061)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7051)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7053)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7054)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7056)	SHA-256 With/without PR; Health test per	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate,	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	section 11.3 of SP 800-90Arev1			and services are available for use.	reseed, generate) health test	
Hash DRBG (A7057)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7058)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7059)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7060)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7061)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7062)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
Hash DRBG (A7063)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A7064)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7051)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7053)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7054)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7056)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7057)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7058)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7059)	HMAC-SHA-256, SHA512 With/without PR;	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate,	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	Health test per section 11.3 of SP 800-90Arev1			and services are available for use.	reseed, generate) health test	
HMAC DRBG (A7060)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7061)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7062)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7063)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
HMAC DRBG (A7064)	HMAC-SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	SP 800-90Ar1 (instantiate, reseed, generate) health test	Module initialization
RSA SigVer (FIPS186-5) (A7051)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Signature verification	Module initialization. Before integrity test.
RSA SigVer (FIPS186-5) (A7062)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Signature verification	Module initialization. Before integrity test.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-5) (A7063)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Signature verification	Module initialization. Before integrity test.
RSA SigVer (FIPS186-5) (A7064)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Signature verification	Module initialization. Before integrity test.
ECDSA SigVer (FIPS186-5) (A7055)	P-256 with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Signature verification	Module initialization
Entropy Source RCT startup	1024 samples	RCT	CAST	Module becomes operational and services are available for use.	Entropy source start-up test	Entropy source initialization
Entropy Source APT startup	1024 samples	APT	CAST	Module becomes operational and services are available for use.	Entropy source start-up test	Entropy source initialization
Entropy Source RCT Continuous	Cutoff C = 61	RCT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
Entropy Source APT Continuous	Cutoff C = 355	APT	CAST	Entropy source is operational	Entropy source continuous test	Continuously

Table 21: Conditional Self-Tests

When all of the pre-operational self-tests pass successfully, the module automatically performs all cryptographic algorithm self-tests (CASTs) as specified in Table “Conditional Self-Tests”. Only if these CASTs also passed successfully, the module transitions to the operational state. No operator intervention is required to reach this point. Services are not available, and data output (via the data output interface) is inhibited during the self-tests. If any of these tests fails, the module transitions to the error state.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A7064)	Message Authentication	SW/FW Integrity	On demand	Manually
RSA SigVer (FIPS186-5) (A7051)	Signature Verification	SW/FW Integrity	On demand	Manually

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-224 (A7051)	KAT	CAST	On demand	Manually
SHA2-224 (A7062)	KAT	CAST	On demand	Manually
SHA2-224 (A7063)	KAT	CAST	On demand	Manually
SHA2-224 (A7064)	KAT	CAST	On demand	Manually
SHA2-256 (A7051)	KAT	CAST	On demand	Manually
SHA2-256 (A7062)	KAT	CAST	On demand	Manually
SHA2-256 (A7063)	KAT	CAST	On demand	Manually
SHA2-256 (A7064)	KAT	CAST	On demand	Manually
SHA2-384 (A7051)	KAT	CAST	On demand	Manually
SHA2-384 (A7062)	KAT	CAST	On demand	Manually
SHA2-384 (A7063)	KAT	CAST	On demand	Manually
SHA2-384 (A7064)	KAT	CAST	On demand	Manually
SHA2-512 (A7051)	KAT	CAST	On demand	Manually
SHA2-512 (A7062)	KAT	CAST	On demand	Manually
SHA2-512 (A7063)	KAT	CAST	On demand	Manually
SHA2-512 (A7064)	KAT	CAST	On demand	Manually
SHA3-224 (A7052)	KAT	CAST	On demand	Manually
SHA3-256 (A7052)	KAT	CAST	On demand	Manually
SHA3-384 (A7052)	KAT	CAST	On demand	Manually
SHA3-512 (A7052)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7051)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7051)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7053)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7053)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7054)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7054)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7056)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB - Decrypt (A7056)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7057)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7057)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7058)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7060)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7060)	KAT	CAST	On demand	Manually
AES-ECB - Encrypt (A7061)	KAT	CAST	On demand	Manually
AES-ECB - Decrypt (A7061)	KAT	CAST	On demand	Manually
AES-CBC - Encrypt (A7051)	KAT	CAST	On demand	Manually
AES-CBC - Decrypt (A7051)	KAT	CAST	On demand	Manually
AES-CBC - Encrypt (A7056)	KAT	CAST	On demand	Manually
AES-CBC - Decrypt (A7056)	KAT	CAST	On demand	Manually
AES-CBC - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-CBC - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-CFB128 - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-CFB128 - Decrypt (A7051)	KAT	CAST	On demand	Manually
AES-CFB128 - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-CBC-CS3 - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-CBC-CS3 - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-OFB - Encrypt (A7059)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-OFB - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7051)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7051)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7053)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7053)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7054)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7054)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7056)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7056)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7057)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7057)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7058)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7058)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7060)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7060)	KAT	CAST	On demand	Manually
AES-GCM - Encrypt (A7061)	KAT	CAST	On demand	Manually
AES-GCM - Decrypt (A7061)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 - Encrypt (A7051)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 - Decrypt (A7051)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-XTS Testing Revision 2.0 - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-CMAC (A7059)	KAT	CAST	On demand	Manually
AES-CTR - Encrypt (A7051)	KAT	CAST	On demand	Manually
AES-CTR - Decrypt (A7051)	KAT	CAST	On demand	Manually
AES-CTR - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-CTR - Decrypt (A7059)	KAT	CAST	On demand	Manually
AES-CCM - Encrypt (A7059)	KAT	CAST	On demand	Manually
AES-CCM - Decrypt (A7059)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A7051)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A7062)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A7063)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A7064)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A7051)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A7062)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A7063)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A7064)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A7051)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A7062)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A7063)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A7064)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-384 (A7051)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A7062)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A7063)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A7064)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A7051)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A7062)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A7063)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A7064)	KAT	CAST	On demand	Manually
HMAC-SHA3-224 (A7052)	KAT	CAST	On demand	Manually
HMAC-SHA3-256 (A7052)	KAT	CAST	On demand	Manually
HMAC-SHA3-384 (A7052)	KAT	CAST	On demand	Manually
HMAC-SHA3-512 (A7052)	KAT	CAST	On demand	Manually
Counter DRBG (A7051)	KAT	CAST	On demand	Manually
Counter DRBG (A7053)	KAT	CAST	On demand	Manually
Counter DRBG (A7054)	KAT	CAST	On demand	Manually
Counter DRBG (A7056)	KAT	CAST	On demand	Manually
Counter DRBG (A7057)	KAT	CAST	On demand	Manually
Counter DRBG (A7058)	KAT	CAST	On demand	Manually
Counter DRBG (A7059)	KAT	CAST	On demand	Manually
Counter DRBG (A7060)	KAT	CAST	On demand	Manually
Counter DRBG (A7061)	KAT	CAST	On demand	Manually
Hash DRBG (A7051)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Hash DRBG (A7053)	KAT	CAST	On demand	Manually
Hash DRBG (A7054)	KAT	CAST	On demand	Manually
Hash DRBG (A7056)	KAT	CAST	On demand	Manually
Hash DRBG (A7057)	KAT	CAST	On demand	Manually
Hash DRBG (A7058)	KAT	CAST	On demand	Manually
Hash DRBG (A7059)	KAT	CAST	On demand	Manually
Hash DRBG (A7060)	KAT	CAST	On demand	Manually
Hash DRBG (A7061)	KAT	CAST	On demand	Manually
Hash DRBG (A7062)	KAT	CAST	On demand	Manually
Hash DRBG (A7063)	KAT	CAST	On demand	Manually
Hash DRBG (A7064)	KAT	CAST	On demand	Manually
HMAC DRBG (A7051)	KAT	CAST	On demand	Manually
HMAC DRBG (A7053)	KAT	CAST	On demand	Manually
HMAC DRBG (A7054)	KAT	CAST	On demand	Manually
HMAC DRBG (A7056)	KAT	CAST	On demand	Manually
HMAC DRBG (A7057)	KAT	CAST	On demand	Manually
HMAC DRBG (A7058)	KAT	CAST	On demand	Manually
HMAC DRBG (A7059)	KAT	CAST	On demand	Manually
HMAC DRBG (A7060)	KAT	CAST	On demand	Manually
HMAC DRBG (A7061)	KAT	CAST	On demand	Manually
HMAC DRBG (A7062)	KAT	CAST	On demand	Manually
HMAC DRBG (A7063)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A7064)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A7051)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A7062)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A7063)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A7064)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-5) (A7055)	KAT	CAST	On demand	Manually
Entropy Source RCT startup	RCT	CAST	On demand	Manually
Entropy Source APT startup	APT	CAST	On demand	Manually
Entropy Source RCT Continuous	RCT	CAST	On demand	Manually
Entropy Source APT Continuous	APT	CAST	On demand	Manually

Table 23: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The Linux kernel immediately stops executing	Software integrity test failure CAST failure	Restart of the module	Kernel Panic

Table 24: Error States

In the error state, the output interface is inhibited, and the module accepts no more inputs or requests (as the module is no longer running).

10.5 Operator Initiation of Self-Tests

The software integrity tests, cryptographic algorithm self-tests, and entropy source start-up tests can be invoked on demand by unloading and subsequently re-initializing the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the AlmaLinux OS 9.6 package in the form of the kernel-5.14.0-570.62.1.el9_6.tuxcare.5, libkcapi-1.4.0-2.el9_6.tuxcare.1, and libkcapi-hmaccalc-1.4.0-2.el9_6.tuxcare.1 RPM packages.
Before the packages are installed, the AlmaLinux OS 9.6 system must operate in approved mode. This can be achieved by:

- Starting the installation in approved mode. Add the fips=1 option to the kernel command line during the system installation. During the software selection stage, do not install any third-party software.
- Switching the system into approved mode after the installation. Execute the *fips-mode-setup --enable* command. Restart the system.

In both cases, the Crypto Officer must verify the AlmaLinux OS 9.6 system operates in approved mode by executing the *fips-mode-setup --check* command, which should output "FIPS mode is enabled."

After installation of the kernel-5.14.0-570.62.1.el9_6.tuxcare.5, libkcapi-1.4.0-2.el9_6.tuxcare.1, and libkcapi-hmaccalc-1.4.0-2.el9_6.tuxcare.1 RPM packages, the Crypto Officer must execute the *cat /proc/sys/crypto/fips_name* command. The Crypto Officer must ensure that the proper name is listed in the output as follows:

TuxCare Kernel Crypto API

Then, the Crypto Officer must execute the *cat /proc/sys/crypto/fips_version* and *rpm -q libkcapi* commands. These commands must output the following (one line per output) depending on the platform in which are executed:

Intel:

5.14.0-570.62.1.el9_6.tuxcare.5.x86_64
libkcapi-1.4.0-2.el9_6.tuxcare.1.x86_64

11.2 Administrator Guidance

The cryptographic boundary consists only of those APIs provided by the Kernel crypto API. If any other API in the Linux kernel is invoked, the user is not interacting with the module specified in this Security Policy.

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 Design and Rules

This section is not applicable.

11.5 Maintenance Requirements

This section is not applicable.

11.6 End of Life

Secure disposal is the customer's responsibility, since the module goes EOL with the operating system.

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the kernel-5.14.0-570.62.1.el9_6.tuxcare.5, libkcapi-1.4.0-2.el9_6.tuxcare.1, and libkcapi-hmaccalc-1.4.0-2.el9_6.tuxcare.1 RPM packages (for Intel platform) can be uninstalled from the AlmaLinux OS 9.6 system.

12 Mitigation of Other Attacks

The module does not offer mitigation of other attacks and therefore this section is not applicable.

Appendix A. Glossary and abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ESV	Entropy Source Validation
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IPsec	Internet Protocol Security
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SSP	Sensitive Security Parameter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS 140-3	FIPS PUB 140-3 - Security Requirements For Cryptographic Modules March 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG	Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	Secure Hash Standard (SHS) March 2012 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS 186-4	Digital Signature Standard (DSS) July 2013 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS 197	Advanced Encryption Standard November 2001 https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS 198-1	The Keyed Hash Message Authentication Code (HMAC) July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 February 2003 https://www.ietf.org/rfc/rfc3447.txt
SP 800-38A	Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001 https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP 800-38A Addendum	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode October 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf
SP 800-38B	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication May 2005 https://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

SP 800-38C	Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality May 2004 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP 800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC November 2007 https://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP 800-38E	Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices January 2010 https://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP 800-90Ar1	Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) June 2005 https://datatracker.ietf.org/doc/html/rfc4106