Juniper Networks, Inc.
Junos® OS Evolved Kernel Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.3

Last update: 2025-01-06

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

Prepared for:
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

Table of Contents

1 General 5
1.1 Overview 5
1.2 Security Levels 5
1.3 Additional Information 5
2 Cryptographic Module Specification 6
2.1 Description 6
2.2 Tested and Vendor Affirmed Module Version and Identification 7
2.3 Excluded Components 8
2.4 Modes of Operation 8
2.5 Algorithms 8
2.6 Security Function Implementations 11
2.7 Algorithm Specific Information 13
2.7.1 AES XTS 13
2.8 RBG and Entropy 13
2.9 Key Generation 14
2.10 Key Establishment 14
2.11 Industry Protocols 14
3 Cryptographic Module Interfaces 15
3.1 Ports and Interfaces 15
4 Roles, Services, and Authentication 16
4.1 Authentication Methods 16
4.2 Roles 16
4.3 Approved Services 16
4.4 Non-Approved Services 19
5 Software/Firmware Security 20
5.1 Integrity Techniques 20
5.2 Initiate on Demand 20
6 Operational Environment 21
6.1 Operational Environment Type and Requirements 21
6.2 Configuration Settings and Restrictions 21
7 Physical Security 22
8 Non-Invasive Security 23
9 Sensitive Security Parameters Management 24

9.1 Storage Areas 24
9.2 SSP Input-Output Methods 24
9.3 SSP Zeroization Methods 24
9.4 SSPs 25
9.5 Transitions 27
10 Self-Tests 28
10.1 Pre-Operational Self-Tests 28
10.2 Conditional Self-Tests 28
10.3 Periodic Self-Test Information 36
10.4 Error States 40
10.5 Operator Initiation of Self-Tests 41
11 Life-Cycle Assurance 42
11.1 Installation, Initialization, and Startup Procedures 42
11.2 Administrator Guidance 42
11.3 Non-Administrator Guidance 42
11.4 End of Life 43
12 Mitigation of Other Attacks 44
Appendix A. Glossary and Abbreviations 45
Appendix B. References 46

List of Tables

Table 1: Security Levels 5
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 7
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 8
Table 4: Modes List and Description 8
Table 5: Approved Algorithms - 10
Table 6: Approved Algorithms - [EVM] 11
Table 7: Non-Approved, Not Allowed Algorithms 11
Table 8: Security Function Implementations 13
Table 9: Entropy Certificates 13
Table 10: Entropy Sources 13
Table 11: Ports and Interfaces 15
Table 12: Roles 16
Table 13: Approved Services 19
Table 14: Non-Approved Services 19
Table 15: Storage Areas 24
Table 16: SSP Input-Output Methods 24
Table 17: SSP Zeroization Methods 25
Table 18: SSP Table 1 26
Table 19: SSP Table 2 27
Table 20: Pre-Operational Self-Tests 28
Table 21: Conditional Self-Tests 36
Table 22: Pre-Operational Periodic Information 37
Table 23: Conditional Periodic Information 40
Table 24: Error States 41

List of Figures

Figure 1: Block Diagram 7

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 2.0 of the Junos® OS Evolved 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

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 Junos® OS Evolved Kernel Cryptographic Module (hereafter referred to as "the module") is a software module running as part of the operating system kernel that provides general purpose cryptographic services.

The module also uses the Junos® OS Evolved OpenSSL Cryptographic Module Version 3.0.8 as a bound module (also referred to as "the bound OpenSSL module") for providing the HMAC-SHA2-256 service used for integrity tests. Sections of this Security Policy which refer to information from the bound module, also known as the Existing Validated Module (or EVM) are marked by [EVM] as per IG 1.A Resolution 5. The bound module is validated to FIPS 140-3 under Cert #5400.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the kernel binary and the fips_chk_hmac binary, which verifies the integrity of the static kernel binary using the bound OpenSSL module HMAC service. In addition, the cryptographic boundary contains the .hmac files which store the expected integrity values for each of the software components. The cryptographic boundary is indicated by the small bold border in Figure 1.

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 TOEPP is indicated by the large thin border in Figure 1.

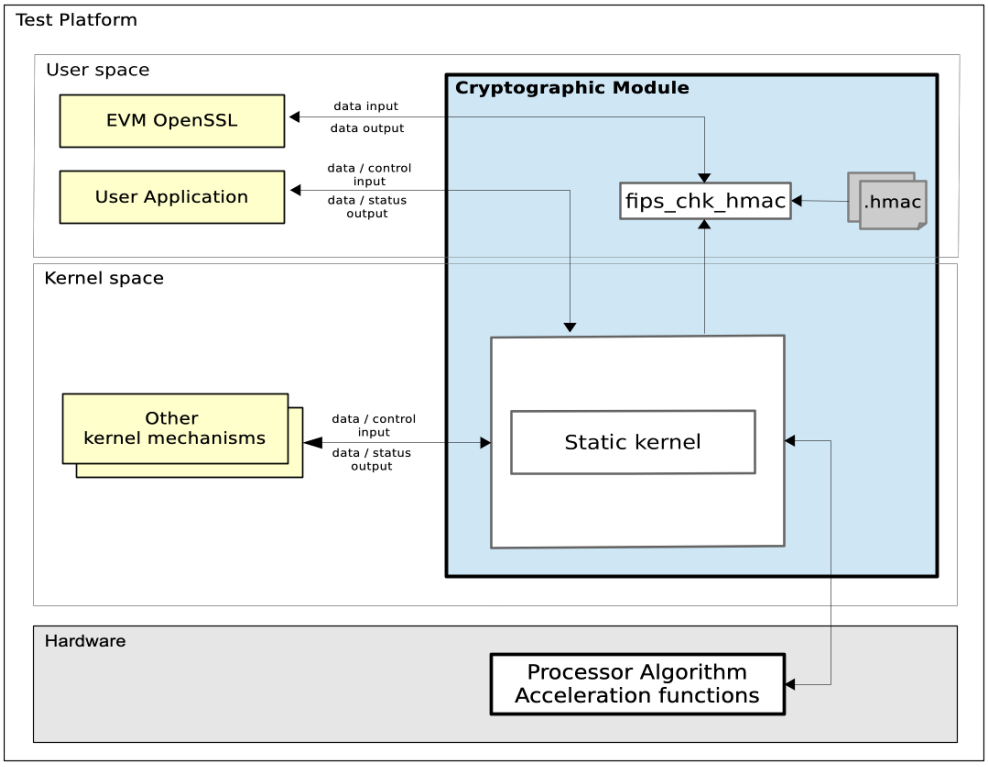

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
/soft/current/bzImage-re-64b.bin, /soft/current/.bzImage-re-64b.bin.hmac, /usr/bin/fips_chk_hmac, and /usr/bin/.fips_chk_hmac.hmac on Juniper Networks® Packet Transport Router Model PTX10001-36MR with Intel® Xeon® D-2163IT	2.0	N/A	HMAC-SHA2-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Junos OS Evolved version 22.4	Juniper Networks® Packet Transport Router Model PTX10001-36MR	Intel® Xeon® D-2163IT	Yes	N/A	2.0
Junos OS Evolved version 22.4	Juniper Networks® Packet Transport Router Model PTX10001-36MR	Intel® Xeon® D-2163IT	No	N/A	2.0

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

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 mode	Automatically entered whenever an approved service is requested	Approved	Mapped to approved service indicator in Section 4.3: respective approved service function returns indicator 0.
Non-approved mode	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 4: 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.

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	A3599, A3600, A3601, A3602	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CMAC	A3599, A3601, A3602	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 8-524288 Increment 8	SP 800-38B
AES-CTR	A3599, A3600, A3601, A3602	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A3599, A3600, A3601, A3602	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A3599, A3600, A3601, A3602	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
Counter DRBG	A3599, A3600, A3601, A3602	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0 Entropy Input - Entropy Input: 128, Entropy Input: 192, Entropy Input: 256 Nonce - Nonce: 128, Nonce: 64 Personalization String Length - Personalization String Length: 0 Returned Bits - 1024, 4096, 512	SP 800-90A Rev. 1
Hash DRBG	A3599, A3600, A3601, A3603, A3604, A3605	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 160, Entropy Input: 256 Nonce - Nonce: 160, Nonce: 256 Personalization String Length - Personalization String Length: 0, 160, Personalization String Length: 0, 256 Additional Input - Additional Input: 0, 160, Additional Input: 0, 256 Returned Bits - 1024, 2048, 320	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
HMAC DRBG	A3599, A3600, A3601, A3603, A3604, A3605	Prediction Resistance - No, Yes Supports Reseed - Yes Mode - SHA-1, SHA2-256, SHA2-512 Entropy Input - Entropy Input: 160, Entropy Input: 256 Nonce - Nonce: 160, Nonce: 256 Personalization String Length - Personalization String Length: 0, 160, Personalization String Length: 0, 256 Additional Input - Additional Input: 0, 160, Additional Input: 0, 256 Returned Bits - 1024, 2048, 320	SP 800-90A Rev. 1
HMAC-SHA-1	A3599, A3603, A3604, A3605	MAC - MAC: 160 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3599, A3603, A3604, A3605	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3599, A3603, A3604, A3605	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3599, A3603, A3604, A3605	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3599, A3603, A3604, A3605	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
SHA-1	A3599, A3603, A3604, A3605	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-224	A3599, A3603, A3604, A3605	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-256	A3599, A3603, A3604, A3605	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-384	A3599, A3603, A3604, A3605	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-512	A3599, A3603, A3604, A3605	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4

Table 5: Approved Algorithms -

[EVM]

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-256	A4246	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Table 6: Approved Algorithms - [EVM]

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES-GCM	Authenticated Encryption and Decryption (not tested by CAVP)
RSA Primitives	RSA Encryption and Decryption primitives (not compliant to SP 800-56Br2)
RSA SigVer	RSA Signature Verification (not tested by CAVP)
RSA with PKCS#1 v1.5	RSA Signature Generation and Signature Verification primitives with PKCS#1 v1.5 padding (not tested by CAVP)

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption	BC-UnAuth	Symmetric encryption		AES-CBC: (A3599, A3600, A3601, A3602) AES-CTR: (A3599, A3600, A3601, A3602) AES-ECB: (A3599, A3600, A3601, A3602) AES-XTS Testing Revision 2.0: (A3599, A3600, A3601, A3602)
Symmetric decryption	BC-UnAuth	Symmetric decryption		AES-CBC: (A3599, A3600, A3601, A3602) AES-CTR: (A3599, A3600, A3601, A3602) AES-ECB: (A3599, A3600, A3601, A3602) AES-XTS Testing Revision 2.0: (A3599, A3600, A3601, A3602)

Name	Type	Description	Properties	Algorithms
Message authentication	MAC	Message authentication		AES-CMAC: (A3599, A3601, A3602) HMAC-SHA-1: (A3599, A3603, A3604, A3605) HMAC-SHA2-224: (A3599, A3603, A3604, A3605) HMAC-SHA2-256: (A3599, A3603, A3604, A3605) HMAC-SHA2-384: (A3599, A3603, A3604, A3605) HMAC-SHA2-512: (A3599, A3603, A3604, A3605)
Random number generation	DRBG	Random number generation		Counter DRBG: (A3599, A3600, A3601, A3602) HMAC DRBG: (A3599, A3600, A3601, A3603, A3604, A3605) Hash DRBG: (A3599, A3600, A3601, A3603, A3604, A3605)
Message digest	SHA	Message digest		SHA-1: (A3599, A3603, A3604, A3605) SHA2-224: (A3599, A3603, A3604, A3605) SHA2-256: (A3599, A3603, A3604, A3605) SHA2-384: (A3599, A3603, A3604, A3605) SHA2-512: (A3599, A3603, A3604, A3605)

Name	Type	Description	Properties	Algorithms
Message authentication for module integrity	MAC	[EVM] Message authentication used to verify the integrity of the module		HMAC-SHA2-256: (A4246)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES XTS

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.

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 two component AES keys used in AES XTS mode shall be generated independently according to the rules for component symmetric keys from SP 800-133 Rev. 2 Section 6.3. The resultant key generated from the two AES keys is formed through concatenation in compliance with SP 800-133 Rev. 2 Section 6.3. To meet the requirements stated in IG C.I, the module implements a check to ensure that the two AES keys input to the module are not identical.

2.8 RBG and Entropy

Cert Number	Vendor Name
E50	Juniper Networks, Inc.

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Kernel CPU Time Jitter RNG version 2.2.0	Non-Physical	See OE Table	64 bits	59.76 bits	N/A

Table 10: Entropy Sources

The module employs Deterministic Random Bit Generators (DRBGs) based on SP 800-90A Rev. 1 for random number generation. The module supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG mechanisms. By default, the module loads the HMAC_DRBG mechanism with SHA2-256 without prediction resistance. When instantiated, these DRBGs can be used to generate random numbers for external usage.

The module uses the Kernel CPU Time Jitter RNG as an entropy source to seed the DRBG. The entropy source is SP 800-90B compliant and is implemented within the module. It resides within the TOEPP and complies with IG 9.3.A Scenario 1b.

The DRBG is seeded with 384 bits of seed material (corresponding to 358 bits of entropy) obtained from the entropy source. During reseeding, the DRBG obtains 256 bits of seed material (corresponding to 239 bits of entropy). The module generates random strings whose strengths are modified by available entropy.

2.9 Key Generation

The module does not provide key generation.

2.10 Key Establishment

The module does not provide key establishment.

2.11 Industry Protocols

The module does not claim cipher suites in compliance to 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 from kernel system calls, AF_ALG type socket.
N/A	Data Output	API output parameters from kernel system calls, AF_ALG type socket.
N/A	Control Input	API function calls, API input parameters for control from kernel system calls, AF_ALG type socket, kernel command line.
N/A	Status Output	API return codes, AF_ALG type socket, kernel logs.

Table 11: Ports and Interfaces

The logical interfaces are the API through which kernel components request services, and the 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.

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	Perform AES Encryption	crypto_skcipher_setkey returns 0	Plaintext, AES key, IV	Ciphertext	Symmetric encryption	Crypto Officer - AES key: W,E
Symmetric Decryption	Perform AES Decryption	crypto_skcipher_setkey returns 0	Ciphertext, AES key, IV	Plaintext	Symmetric decryption	Crypto Officer - AES key: W,E
Random Number Generation	Generate random numbers	crypto_rng_get_bytes returns 0	Output length	Random bytes	Random number generation	Crypto Officer - DRBG entropy input string: G,W,E - Hash_DRBG seed: G,E - HMAC_DRBG seed: G,E - CTR_DRBG seed: G,E - HMAC_DRB

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G internal state (V, Key): G,W,E - CTR_DRBG internal state (V, Key): G,W,E - Hash_DRBG internal state (V, C): G,W,E
Message Digest	Compute SHA hashes	crypto_shash_init returns 0	Message	Digest	Message digest	Crypto Officer
Message Authentication	Compute HMAC or CMAC	crypto_shash_init returns 0	Message, HMAC or AES key	MAC tag	Message authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Encrypt then MAC	Encrypt plaintext with AES (CBC or CTR) and use HMAC to create tag	crypto_shash_init returns 0	Plaintext, AES key, HMAC key, IV	Ciphertext, MAC tag	Symmetric encryption Message authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Decrypt then Verify	Decrypt a ciphertext using AES (CBC or CTR) and use HMAC to verify tag	crypto_shash_init returns 0	Ciphertext, AES key, HMAC key, IV, MAC tag	Plaintext or failure	Symmetric decryption Message authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Error Detection Code	Compute an EDC (crc32c, crct10dif)	None	Data	EDC	None	Crypto Officer
Memory Copy Operation	Copy memory	None	Source location, destination location	N/A	None	Crypto Officer
Generic System Call	Use the kernel to	None	[various]	[various]	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	perform various non-cryptographic operations					
Show Status	Return the module status	None	N/A	Module status	None	Crypto Officer
Self-Test	Perform the CASTs and the integrity test	None	None	Pass/fail results of tests	Symmetric encryption Symmetric decryption Message authentication Random number generation Message digest Message authentication for module integrity	Crypto Officer
Zeroization	Zeroize all SSPs	None	SSPs to be zeroized	None	None	Crypto Officer - AES key: Z - HMAC key: Z - DRBG entropy input string: Z - Hash_DRBG internal state (V, C): Z - HMAC_DRBG internal state (V, Key): Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						CTR_DRBG internal state (V, Key): Z - HMAC_DRBG seed: Z - Hash_DRBG seed: Z - CTR_DRBG seed: Z
Show Version	Return module name and version information	None	None	Module name, module version	None	Crypto Officer

Table 13: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES-GCM	Authenticated Encryption and Decryption	AES-GCM	CO
RSA Encrypt/Decrypt	RSA Encryption and Decryption primitives	RSA Primitives	CO
RSA SigVer	RSA Signature Verification	RSA SigVer	CO
RSA PKCS#1 v1.5	Signature Generation and Signature Verification primitives with PKCS#1 v1.5 padding	RSA with PKCS#1 v1.5	CO

Table 14: Non-Approved Services

5 Software/Firmware Security

5.1 Integrity Techniques

The module verifies its integrity through the following mechanisms:

- The integrity of the static kernel binary is ensured with the HMAC-SHA2-256 value stored in the corresponding .hmac file that is computed at kernel build time. During Pre-Operational Self-Tests, the module invokes the fips_chk_hmac utility to calculate the HMAC value of the static kernel binary file (relying on the HMAC service provided by the bound OpenSSL module), and then compares it with the pre-stored one. If the two HMAC values do not match, the kernel panics to indicate that the test fails and the module enters the error state.
- The integrity of the fips_chk_hmac utility itself is performed before the integrity tests of the static kernel binary, and ensured with the HMAC-SHA2-256 value stored in the corresponding .hmac file that is computed at the utility build time. The utility makes use of OpenSSL's HMAC service to calculate the HMAC value, and then compares it with the pre-stored one. If the two HMAC values do not match, the kernel panics to indicate that the test fails and the module enters the error state.

The HMAC key is stored within the fips_chk_hmac utility binary.

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 operating system provides process isolation and memory protection mechanisms that ensure appropriate separation for memory access among the processes on the system. Each process has control over its own data and uncontrolled access to the data of other processes is prevented.

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 environments. 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	Operator calling application (TOEPP)	Cryptographic module	Plaintext	Manual	Electronic	

Table 16: SSP Input-Output Methods

The module does not output SSPs.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Wipe and Free memory block allocated	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 cipher related zeroization API function: crypto_free_cipher(), crypto_free_skcipher(), crypto_free_aead() for AES keys; crypto_free_shash(), crypto_free_ahash() for HMAC keys; crypto_free_rng() for DRBG SSPs
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
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten	By unloading and reloading the module

Zeroization Method	Description	Rationale	Operator Initiation
		within nanoseconds when power is removed.	

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	128, 196, 256 bits - 128, 196, 256 bits	Symmetric Key - CSP			Symmetric encryption Symmetric decryption Message authentication
HMAC key	HMAC key	112-524288 bits - 112-256 bits	Symmetric Key - CSP			Message authentication
DRBG entropy input string	Entropy input string (IG D.L compliant)	128-384 bits - 119-358 bits	Entropy Input - CSP	Random number generation		Random number generation
Hash_DRBG internal state (V, C)	Internal state of DRBG (IG D.L compliant)	880, 1776 bits - 128, 256 bits	Internal State - CSP	Random number generation		Random number generation
HMAC_DRBG internal state (V, Key)	Internal state of DRBG (IG D.L compliant)	320, 512, 1024 bits - 128, 256 bits	Internal State - CSP	Random number generation		Random number generation
CTR_DRBG internal state (V, Key)	Internal state of DRBG (IG D.L compliant)	256, 320, 384 bits - 128, 192, 256 bits	Internal State - CSP	Random number generation		Random number generation
Hash_DRBG seed	DRBG seed derived from entropy input (IG D.L compliant)	440, 888 bits - 128, 256 bits	Seed - CSP	Random number generation		Random number generation
HMAC_DRBG seed	DRBG seed derived from entropy input (IG D.L compliant)	160, 256, 512 bits - 128, 256 bits	Seed - CSP	Random number generation		Random number generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
CTR_DRBG seed	DRBG seed derived from entropy input (IG D.L compliant)	256, 320, 384 bits - 128, 192, 256 bits	Seed - CSP	Random number generation		Random number generation

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
HMAC key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
DRBG entropy input string		RAM:Plaintext	From generation until DRBG seed is created	Wipe and Free memory block allocated Automatic Module Reset	CTR_DRBG seed:Derives HMAC_DRBG seed:Derives Hash_DRBG seed:Derives
Hash_DRBG internal state (V, C)		RAM:Plaintext	From DRBG instantiation to DRBG termination	Wipe and Free memory block allocated Automatic Module Reset	Hash_DRBG seed:Derived From
HMAC_DRBG internal state (V, Key)		RAM:Plaintext	From DRBG instantiation to DRBG termination	Wipe and Free memory block allocated Automatic Module Reset	HMAC_DRBG seed:Derived From
CTR_DRBG internal state (V, Key)		RAM:Plaintext	From DRBG instantiation to DRBG termination	Wipe and Free memory block allocated Automatic Module Reset	CTR_DRBG seed:Derived From
Hash_DRBG seed		RAM:Plaintext	While DRBG is instantiated	Wipe and Free memory block allocated Automatic Module Reset	DRBG entropy input string:Derived From Hash_DRBG

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					internal state (V, C):Derives
HMAC_DRBG seed		RAM:Plaintext	While DRBG is instantiated	Wipe and Free memory block allocated Automatic Module Reset	DRBG entropy input string:Derived From HMAC_DRBG internal state (V, Key):Derives
CTR_DRBG seed		RAM:Plaintext	While DRBG is instantiated	Wipe and Free memory block allocated Automatic Module Reset	DRBG entropy input string:Derived From CTR_DRBG internal state (V, Key):Derives

Table 19: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes except signature verification 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-SHA2-256 (A4246) - kernel	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	[EVM] Integrity test for static kernel binary
HMAC-SHA2-256 (A4246) - fips_chk_hmac	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	[EVM] Integrity test for fips_chk_hmac binary

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. The algorithms used for the integrity test (i.e., HMAC-SHA2-256) run their CASTs before the integrity test is performed. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the pre-operational software integrity self-tests are successfully completed. The module transitions to the operational state only after the pre-operational self-tests are passed successfully. If any pre-operational self-test fails, the module will instead transition to the Error State.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A3599) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3600) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3601) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3602) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3599) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
AES-CBC (A3600) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3601) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3602) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3599) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3600) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3601) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3602) - Encryption	128, 192, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3599) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3600) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3601) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A3602) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
AES-CBC (A3599) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3600) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3601) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A3602) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3599) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3600) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3601) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CTR (A3602) - Decryption	128, 192, 256-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CMAC (A3599) - Encryption	128, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A3601) - Encryption	128, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A3602) - Encryption	128, 256-bit keys, encrypt	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
SHA-1 (A3599)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3603)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3604)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A3605)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3599)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3603)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3604)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A3605)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3599)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3603)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A3604)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
SHA2-256 (A3605)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3599)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3603)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3604)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A3605)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3599)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3603)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3604)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A3605)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3599)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3603)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
HMAC-SHA-1 (A3604)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A3605)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3599)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3603)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3604)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A3605)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3599)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3603)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3604)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A3605)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3599)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
HMAC-SHA2-384 (A3603)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3604)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A3605)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3599)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3603)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3604)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A3605)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
Counter DRBG (A3599)	AES-128, AES-192, AES-256, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Counter DRBG (A3600)	AES-128, AES-192, AES-256, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Counter DRBG (A3601)	AES-128, AES-192, AES-256, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Counter DRBG (A3602)	AES-128, AES-192, AES-256,	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	with and without PR					before the integrity test
HMAC DRBG (A3599)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
HMAC DRBG (A3600)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
HMAC DRBG (A3601)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
HMAC DRBG (A3603)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
HMAC DRBG (A3604)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
HMAC DRBG (A3605)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Hash DRBG (A3599)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Hash DRBG (A3600)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Hash DRBG (A3601)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A3603)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Hash DRBG (A3604)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Hash DRBG (A3605)	SHA-1, SHA2-256, SHA2-512, with and without PR	KAT	CAST	Module becomes operational	Compliant with SP 800-90A Rev. 1	Test runs at power-on before the integrity test
Entropy Source Health Tests - Startup RCT	Cutoff value = 31	RCT	CAST	Module becomes operational	RCT performed on 1024 noise source samples	Test runs at startup as defined in SP 800-90B
Entropy Source Health Tests - Startup APT	Cutoff value = 325, window size = 512	APT	CAST	Module becomes operational	APT performed on 1024 noise source samples	Test runs at startup as defined in SP 800-90B
Entropy Source Health Tests - Continuous RCT	Cutoff value = 31	RCT	CAST	Module remains operational	RCT performed continuously	Test runs continuously as defined in SP 800-90B
Entropy Source Health Tests - Continuous APT	Cutoff value = 325, window size = 512	APT	CAST	Module remains operational	APT performed continuously	Test runs continuously as defined in SP 800-90B
HMAC-SHA2-256 (A4246)	SHA2-256	KAT	CAST	Module becomes operational	[EVM]	Test runs at power-on before the integrity test

Table 21: Conditional Self-Tests

The module performs self-tests on all approved cryptographic algorithms as part of the approved services supported in the approved mode of operation, using the tests shown in the table above. Services are not available, and data output (via the data output interface) is inhibited during the conditional 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-SHA2-256 (A4246) - kernel	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A4246) - fips_chk_hmac	MAC tag verification	SW/FW Integrity	On Demand	Manually

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A3599) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3600) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3601) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3602) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A3599) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A3600) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A3601) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A3602) - Encryption	KAT	CAST	On Demand	Manually
AES-CTR (A3599) - Encryption	KAT	CAST	On Demand	Manually
AES-CTR (A3600) - Encryption	KAT	CAST	On Demand	Manually
AES-CTR (A3601) - Encryption	KAT	CAST	On Demand	Manually
AES-CTR (A3602) - Encryption	KAT	CAST	On Demand	Manually
AES-ECB (A3599) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3600) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3601) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A3602) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A3599) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A3600) - Decryption	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A3601) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A3602) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A3599) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A3600) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A3601) - Decryption	KAT	CAST	On Demand	Manually
AES-CTR (A3602) - Decryption	KAT	CAST	On Demand	Manually
AES-CMAC (A3599) - Encryption	KAT	CAST	On Demand	Manually
AES-CMAC (A3601) - Encryption	KAT	CAST	On Demand	Manually
AES-CMAC (A3602) - Encryption	KAT	CAST	On Demand	Manually
SHA-1 (A3599)	KAT	CAST	On Demand	Manually
SHA-1 (A3603)	KAT	CAST	On Demand	Manually
SHA-1 (A3604)	KAT	CAST	On Demand	Manually
SHA-1 (A3605)	KAT	CAST	On Demand	Manually
SHA2-224 (A3599)	KAT	CAST	On Demand	Manually
SHA2-224 (A3603)	KAT	CAST	On Demand	Manually
SHA2-224 (A3604)	KAT	CAST	On Demand	Manually
SHA2-224 (A3605)	KAT	CAST	On Demand	Manually
SHA2-256 (A3599)	KAT	CAST	On Demand	Manually
SHA2-256 (A3603)	KAT	CAST	On Demand	Manually
SHA2-256 (A3604)	KAT	CAST	On Demand	Manually
SHA2-256 (A3605)	KAT	CAST	On Demand	Manually
SHA2-384 (A3599)	KAT	CAST	On Demand	Manually
SHA2-384 (A3603)	KAT	CAST	On Demand	Manually
SHA2-384 (A3604)	KAT	CAST	On Demand	Manually
SHA2-384 (A3605)	KAT	CAST	On Demand	Manually
SHA2-512 (A3599)	KAT	CAST	On Demand	Manually
SHA2-512 (A3603)	KAT	CAST	On Demand	Manually
SHA2-512 (A3604)	KAT	CAST	On Demand	Manually
SHA2-512 (A3605)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3603)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA-1 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3605)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3603)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A3605)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3603)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3605)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3603)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A3605)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3603)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3605)	KAT	CAST	On Demand	Manually
Counter DRBG (A3599)	KAT	CAST	On Demand	Manually
Counter DRBG (A3600)	KAT	CAST	On Demand	Manually
Counter DRBG (A3601)	KAT	CAST	On Demand	Manually
Counter DRBG (A3602)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A3599)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3600)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3601)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3603)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3604)	KAT	CAST	On Demand	Manually
HMAC DRBG (A3605)	KAT	CAST	On Demand	Manually
Hash DRBG (A3599)	KAT	CAST	On Demand	Manually
Hash DRBG (A3600)	KAT	CAST	On Demand	Manually
Hash DRBG (A3601)	KAT	CAST	On Demand	Manually
Hash DRBG (A3603)	KAT	CAST	On Demand	Manually
Hash DRBG (A3604)	KAT	CAST	On Demand	Manually
Hash DRBG (A3605)	KAT	CAST	On Demand	Manually
Entropy Source Health Tests - Startup RCT	RCT	CAST	On Demand	Manually
Entropy Source Health Tests - Startup APT	APT	CAST	On Demand	Manually
Entropy Source Health Tests - Continuous RCT	RCT	CAST	On Demand	Manually
Entropy Source Health Tests - Continuous APT	APT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4246)	KAT	CAST	On Demand	Manually

Table 23: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	General-purpose error state	Failure of pre-operational or conditional tests Failure of Entropy source Health Tests	Restart 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).
The error can be recovered by a restart (i.e., powering off and powering on) of the module.
The error can also be recovered by recovering from snapshot of the module as follows:

1. Issue the request system snapshot operational mode command:
   request system snapshot
2. Use the *show system snapshot* operational mode command to see the snapshot images available on the Routing Engines:
   show system snapshot
3. To recover the primary Routing Engine using the snapshot, boot the Routing Engine from the secondary SSD (disk2):
   request node reboot re0 disk2
4. If the Routing Engine has successfully booted from the secondary SSD, after the Routing Engine boots up, you see a message similar to the following before the login prompt:
   WARNING: THIS DEVICE HAS BOOTED FROM ALTERNATE DEVICE (/dev/sdb)

10.5 Operator Initiation of Self-Tests

All self-tests, with the exception of the continuous health 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 pre-installed in the junos-evo-install-ptx-fixed-x86-64-22.4R2.11-S1-EVO.iso image. The procedures on how to mount and install the image are listed in the software-install-and-upgrade-overview-evo documentation[1]. The Crypto Officer shall follow this Security Policy to configure the operational environment and to operate the module as a FIPS 140-3 validated module.

To configure the operating environment to run in the approved mode, the following shall be performed with the root privilege:

1. Enter CLI configuration mode.
2. Configure FIPS level to 1:
   set system fips level 1
3. Commit changes:
   commit
4. Exit configuration mode to enter operational mode:
   exit
5. Reboot the system with the new settings (answer yes to prompt):
   request system reboot

The Crypto Officer should check the existence of the file, */proc/sys/crypto/fips_enabled*, and that it contains "1". If the file does not exist or does not contain "1", the operating environment is not configured to operate properly in the approved mode.

11.2 Administrator Guidance

In order to run in the Approved mode, the module must be operated using the approved services, with their corresponding approved and allowed cryptographic algorithms provided in this Security Policy. In addition, key sizes must comply with [SP 800-131A Rev. 2].
Once the OE is properly configured, the operator is responsible to verify that the installation and configuration is completed. For such purpose, the following command "*cat /proc/sys/fips_version*" must return:
Junos OS Evolved Kernel Cryptographic Module 2.0

The Junos OS Evolved OpenSSL Cryptographic Module version 3.0.8 is a bound module that shall also be installed and configured as described in section 11.1 of its Security Policy. The administrator shall follow the steps to install the module and verify its version.

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

[1] URL: https://www.juniper.net/documentation/us/en/software/junos/junos-install-upgrade-evo/topics/concept/software-install-and-upgrade-overview-evo.html

11.4 End of Life

As a first step for the secure sanitization, the module needs to be powered off which will erase the SSPs in the volatile memory. Then, the files listed related to the static kernel binary and fips_chk_hmac utility must be deleted using the command “shred -zu <file_name>”. Then, for the actual deprecation, the module will be upgraded to a newer version that is approved.

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
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
HMAC	Keyed-Hash Message Authentication Code
KAT	Known Answer Test
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
PAA	Processor Algorithm Acceleration
PKCS	Public-Key Cryptography Standards
RSA	Rivest, Shamir, Adleman
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://doi.org/10.6028/NIST.FIPS.140-3
FIPS 140-3 IG [09-02-2025]	**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
SP 800-38A	**Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://doi.org/10.6028/NIST.SP.800-38A
SP 800-38B	**Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 https://doi.org/10.6028/NIST.SP.800-38B
SP 800-38E	**Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality of Storage Devices** January 2010 https://doi.org/10.6028/NIST.SP.800-38E
FIPS 180-4	**Secure Hash Standard (SHS)** August 2015 https://doi.org/10.6028/NIST.FIPS.180-4
FIPS 198-1	**The Keyed-Hash Message Authentication Code (HMAC)** July 2008 https://doi.org/10.6028/NIST.FIPS.198-1
SP 800-90A Rev. 1	**Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://doi.org/10.6028/NIST.SP.800-90Ar1
SP 800-90B	**Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://doi.org/10.6028/NIST.SP.800-90B
SP 800-131A Rev. 2	**Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://doi.org/10.6028/NIST.SP.800-131Ar2
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** Feburary 2003 https://www.ietf.org/rfc/rfc3447.txt