Juniper Networks, Inc.

Junos OS Evolved Kernel Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy
Document Version: 1.1
Last update: 08-28-2024

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

List of Tables

List of Figures

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 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.

1.1.1 How this Security Policy was prepared

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.

1.2 Security Levels

Section	Security Level
1	1
2	1
3	1
4	1
5	1
6	1
7	N/A
8	N/A
9	1
10	1
11	1
12	N/A

Table 1: Security Levels

1.3 Additional Information [O]

Not applicable.

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. It is bound to the Junos OS Evolved OpenSSL Cryptographic Module Version 3.0.8 validated under FIPS certificate #4775 to check the integrity of its static kernel binary file.

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics:

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.

Tested Operational Environment’s Physical Perimeter (TOEPP) [O]:

The TOEPP of the module is defined as the general-purpose computer on which the module is installed.

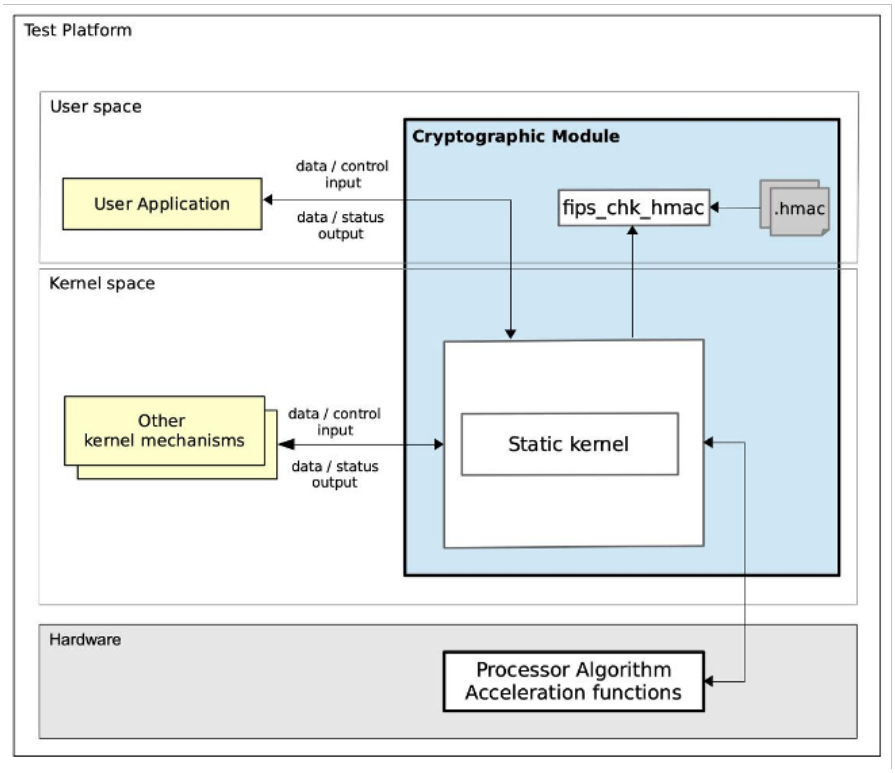

Figure 1: Block Diagram

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
/usr/bin/fips_chk_hmac (makes use of HMAC service from bound OpenSSL module)	2.0	N/A	HMAC-SHA2-256
/soft/current/bzImage-re-64b.bin	2.0	N/A	HMAC-SHA2-256

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

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Junos OS Evolved version 22.4	Juniper Networks® Packet Transport Router Model PTX10001-36MR	Intel(R) Xeon(R) D-2163IT	With and without AES-NI, SHA Extensions (PAA)		2.0

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

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

N/A for this module.

The module is not tested in any vendor-affirmed operational environment.

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	Equivalent to the indicator of the requested service as defined in Section 4.3
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service as defined in Section 4.3

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 [O]:

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.

Degraded Mode Description [O]:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3599	-	SP 800-38A
AES-CMAC	A3599	-	SP 800-38B
AES-CTR	A3599	-	SP 800-38A
AES-ECB	A3599	-	SP 800-38A
AES-XTS Testing Revision 2.0	A3599	-	SP 800-38E
Counter DRBG	A3599	-	SP 800-90A Rev. 1
Hash DRBG	A3599	-	SP 800-90A Rev. 1
HMAC DRBG	A3599	-	SP 800-90A Rev. 1
HMAC-SHA-1	A3599	-	FIPS 198-1
HMAC-SHA2-224	A3599	-	FIPS 198-1
HMAC-SHA2-256	A3599	-	FIPS 198-1
HMAC-SHA2-384	A3599	-	FIPS 198-1
HMAC-SHA2-512	A3599	-	FIPS 198-1
SHA-1	A3599	-	FIPS 180-4
SHA2-224	A3599	-	FIPS 180-4
SHA2-256	A3599	-	FIPS 180-4
SHA2-384	A3599	-	FIPS 180-4
SHA2-512	A3599	-	FIPS 180-4
AES-CBC	A3600	-	SP 800-38A
AES-CTR	A3600	-	SP 800-38A
AES-ECB	A3600	-	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-XTS Testing Revision 2.0	A3600	-	SP 800-38E
Counter DRBG	A3600	-	SP 800-90A Rev. 1
Hash DRBG	A3600	-	SP 800-90A Rev. 1
HMAC DRBG	A3600	-	SP 800-90A Rev. 1
AES-CBC	A3601	-	SP 800-38A
AES-CMAC	A3601	-	SP 800-38B
AES-CTR	A3601	-	SP 800-38A
AES-ECB	A3601	-	SP 800-38A
AES-XTS Testing Revision 2.0	A3601	-	SP 800-38E
Counter DRBG	A3601	-	SP 800-90A Rev. 1
Hash DRBG	A3601	-	SP 800-90A Rev. 1
HMAC DRBG	A3601	-	SP 800-90A Rev. 1
AES-CBC	A3602	-	SP 800-38A
AES-CMAC	A3602	-	SP 800-38B
AES-CTR	A3602	-	SP 800-38A
AES-ECB	A3602	-	SP 800-38A
AES-XTS Testing Revision 2.0	A3602	-	SP 800-38E
Counter DRBG	A3602	-	SP 800-90A Rev. 1
Hash DRBG	A3603	-	SP 800-90A Rev. 1
HMAC DRBG	A3603	-	SP 800-90A Rev. 1
HMAC-SHA-1	A3603	-	FIPS 198-1
HMAC-SHA2-224	A3603	-	FIPS 198-1
HMAC-SHA2-256	A3603	-	FIPS 198-1
HMAC-SHA2-384	A3603	-	FIPS 198-1
HMAC-SHA2-512	A3603	-	FIPS 198-1
SHA-1	A3603	-	FIPS 180-4
SHA2-224	A3603	-	FIPS 180-4
SHA2-256	A3603	-	FIPS 180-4
SHA2-384	A3603	-	FIPS 180-4
SHA2-512	A3603	-	FIPS 180-4
Hash DRBG	A3604	-	SP 800-90A Rev. 1
HMAC DRBG	A3604	-	SP 800-90A Rev. 1
HMAC-SHA-1	A3604	-	FIPS 198-1
HMAC-SHA2-224	A3604	-	FIPS 198-1
HMAC-SHA2-256	A3604	-	FIPS 198-1
HMAC-SHA2-384	A3604	-	FIPS 198-1
HMAC-SHA2-512	A3604	-	FIPS 198-1
SHA-1	A3604	-	FIPS 180-4
SHA2-224	A3604	-	FIPS 180-4
SHA2-256	A3604	-	FIPS 180-4
SHA2-384	A3604	-	FIPS 180-4
SHA2-512	A3604	-	FIPS 180-4
Hash DRBG	A3605	-	SP 800-90A Rev. 1
HMAC DRBG	A3605	-	SP 800-90A Rev. 1
HMAC-SHA-1	A3605	-	FIPS 198-1
HMAC-SHA2-224	A3605	-	FIPS 198-1
HMAC-SHA2-256	A3605	-	FIPS 198-1
HMAC-SHA2-384	A3605	-	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512	A3605	-	FIPS 198-1
SHA-1	A3605	-	FIPS 180-4
SHA2-224	A3605	-	FIPS 180-4
SHA2-256	A3605	-	FIPS 180-4
SHA2-384	A3605	-	FIPS 180-4
SHA2-512	A3605	-	FIPS 180-4
HMAC-SHA2-256	A4249	-	FIPS 198-1
HMAC-SHA2-256	A4246	-	FIPS 198-1
HMAC-SHA2-256	A4247	-	FIPS 198-1
HMAC-SHA2-256	A4248	-	FIPS 198-1

Table 5: 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	Authenticated Encryption and Decryption
RSA	RSA Encryption and Decryption primitives
RSA	RSA Signature Verification
RSA	Signature Generation and Signature Verification primitives with PKCS#1 v1.5 padding

Table 6: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric encryption	BC-UnAuth	Symmetric encryption	AES-CBC:128, 192, 256-bit keys with 128-256 bits key strength AES-CTR:128, 192, 256-bit keys with 128-256 bits key strength AES-ECB:128,	AES-CBC AES-CBC AES-CBC AES-CBC AES-CTR AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB

Name	Type	Description	Properties	Algorithms
			192, 256-bit keys with 128-256 bits key strength AES-XTS Testing Revision 2.0:128, 256-bit keys with 128, 256 bits key strength	AES-ECB AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Message authentication	MAC	Message authentication	AES-CMAC:128, 192, 256-bit keys with 128-256 bits key strength HMAC-SHA-1:112-524288 bit keys with 128-256 bits key strength HMAC-SHA2-224:112-256 bit keys with 128-256 bits key strength HMAC-SHA2-256:112-256 bit keys with 128-256 bits key strength HMAC-SHA2-384:112-256 bit keys with 128-256 bits key strength HMAC-SHA2-512:112-256 bit keys with 128-256 bits key strength	AES-CMAC AES-CMAC AES-CMAC HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512

Name	Type	Description	Properties	Algorithms
Random number generation	DRBG	Random number generation	Counter DRBG:128, 192, 256 bits HMAC DRBG:128, 256 bits Hash DRBG:128, 256 bits	Counter DRBG Counter DRBG Counter DRBG Counter DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG
Message digest	SHA	Message digest	SHA-1:N/A SHA2-224:N/A SHA2-256:N/A SHA2-384:N/A SHA2-512:N/A	SHA-1 SHA-1 SHA-1 SHA-1 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-512 SHA2-512 SHA2-512 SHA2-512
Symmetric decryption	BC-UnAuth	Symmetric decryption	AES-CBC:128, 192, 256-bit keys with 128-256 bits key strength AES-CTR:128, 192, 256-bit keys with 128-256 bits key strength AES-ECB:128, 192, 256-bit keys with 128-256 bits key	AES-CBC AES-CBC AES-CBC AES-CBC AES-CTR AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB AES-ECB AES-XTS Testing Revision 2.0

Name	Type	Description	Properties	Algorithms
			strength AES-XTS Testing Revision 2.0:128, 256-bit keys with 128, 256 bits key strength	AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Authenticated encryption	BC-Auth	Encrypt and authenticate a plaintext	Key size(s):AES-CBC/AES-CTR: 128, 192, 256 bits with 128-256 bits of security strength; HMAC: 112-524288 bit keys with 128-256 bits key strength	AES-CBC AES-CBC AES-CBC AES-CBC AES-CTR AES-CTR AES-CTR AES-CTR HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-512
Authenticated decryption	BC-Auth	Decrypt and authenticate a ciphertext	Key size(s):AES-CBC/AES-CTR: 128, 192, 256 bits with 128-256 bits of security strength; HMAC: 112-524288 bit keys with 128-256 bits key strength	AES-CBC AES-CBC AES-CBC AES-CBC AES-CTR AES-CTR AES-CTR AES-CTR HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512
(OpenSSL) Message authentication	MAC	HMAC-SHA2-256 used in fips_chk_hmac integrity check	Key Size:256-bit Key	HMAC-SHA2-256 HMAC-SHA2-256

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-256 HMAC-SHA2-256

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 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. 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.

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.

2.8 RBG and Entropy

Cert Number	Vendor Name
E50	Juniper Networks, Inc.

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
SP 800-90B compliant ENT(NP) (ESV cert. E50)	Non-Physical	Junos OS Evolved version 22.4 on Juniper Networks® Packet Transport Router Model PTX10001-36MR	64 bits	59.76 bits	Linear-Feedback Shift Register (LFSR)

Table 9: Entropy Sources

The module employs the Deterministic Random Bit Generator (DRBG) based on [SP800-90Arev1] for the random number generation. The DRBG supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG mechanisms. The module obtains an entropy input string from the SP800-90B compliant ENT(NP), whose length depends on each DRBG mechanism, meeting the requirements of SP800-90Arev1 (128 to 384 bits).

The module loads by default the DRBG using 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.

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.

2.12 Additional Information [O]

Not applicable.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs.	Data Input	API input parameters from kernel system calls, AF_ALG type socket
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs.	Data Output	API output parameters from kernel system calls, AF_ALG type socket
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs.	Control Input	API function calls, API input parameters for control from kernel system calls, AF_ALG type socket, kernel command line
As a software-only module, the module does not have physical ports. Physical Ports are interpreted to be the physical ports of the hardware platform on which it runs.	Status Output	API return codes, AF_ALG type socket, kernel logs

Table 10: 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.

3.2 Trusted Channel Specification [O]

The module does not implement a trusted channel.

3.3 Control Interface Not Inhibited [O]

The module does not implement a control output interface.

3.4 Additional Information [O]

Not applicable.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not implement authentication.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 11: 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
Message digest	Compute SHA hashes	crypto_shash_init returns 0	Message	Digest value	Message digest	Crypto Officer
Symmetric encryption	Perform AES encryption	crypto_skcipher_setkey returns 0	AES key, plaintext	Ciphertext	Symmetric encryption	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric decryption	Perform AES decryption	crypto_skcipher_setkey returns 0	AES key, ciphertext	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: W,E - DRBG seed: G,E - DRBG internal state (V, Key): G,W,E - DRBG internal state (V, C): G,W,E
Message authentication	Compute HMAC/AES-based CMAC	crypto_shash_init returns 0	AES: AES key, message; HMAC: HMAC key, message	MAC tag	Message authentication	Crypto Officer - AES key: W,E - HMAC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: W,E
Authenticated encryption	Encrypt and authenticate a plaintext	crypto_aead_setkey returns 0	AES key, plaintext	Ciphertext, MAC tag	Authenticated encryption	Crypto Officer - AES key: W,E - HMAC key: W,E
Authenticated decryption	Decrypt and authenticate a ciphertext	crypto_aead_setkey returns 0	AES key, ciphertext, MAC tag	Plaintext or failure	Authenticated decryption	Crypto Officer - AES key: W,E - HMAC key: W,E
Error detection code	Compute an EDC (crc32c, crct10dif)	None	Message	EDC	None	Crypto Officer
Memory copy operation	Copy operation	None	Source, destination, offset, amount	Return codes and/or log messages	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
Show status	Return the module status	None	N/A	Module status	None	Crypto Officer
Self-tests	Perform the CASTs	None	N/A	Pass/fail	Symmetric encryption	Crypto

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	and the integrity test				Message authentication Random number generation Message digest Symmetric decryption Authenticated encryption Authenticated decryption (OpenSSL) Message authentication	Officer
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - DRBG entropy input string: Z - DRBG internal state (V, Key): Z - DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						internal state (V, C): Z - DRBG seed: Z
Show version	Return the module name and version	None	N/A	Name and version information	None	Crypto Officer

Table 12: Approved Services

The table above 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
Authenticated encryption	Perform AES-GCM encryption	AES-GCM	CO
Authenticated decryption	Perform AES-GCM decryption	AES-GCM	CO
RSA encryption primitive	Compute the raw RSA encryption of a number	RSA	CO
RSA decryption primitive	Compute the raw RSA decryption of a number	RSA	CO
RSA signature generation primitive	Generate a digital signature for a pre-hashed message	RSA	CO
RSA signature verification primitive	Verify a digital signature for a pre-hashed message	RSA	CO
RSA signature verification	Verify RSA-based signature	RSA	CO

Table 13: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

4.6 Bypass Actions and Status [O]

The module does not implement a bypass capability.

4.7 Cryptographic Output Actions and Status [O]

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

4.8 Additional Information [O]

Not applicable.

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.

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.

5.3 Open-Source Parameters [O]

Not applicable.

5.4 Additional Information [O]

Not applicable.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied [O]:

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 [O]

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.

6.3 Additional Information [O]

Not applicable.

7 Physical Security

7.1 Mechanisms and Actions Required [O]

N/A for this module.

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

7.2 User Placed Tamper Seals [O]

Not applicable.

7.3 Filler Panels [O]

Not applicable.

7.4 Fault Induction Mitigation [O]

Not applicable.

7.5 EFP/EFT Information [O]

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature			
HighTemperature			
LowVoltage			
HighVoltage			

Table 14: EFP/EFT Information

Not applicable.

7.6 Hardness Testing Temperature Ranges [O]

Temperature Type	Temperature
LowTemperature	
HighTemperature	

Table 15: Hardness Testing Temperatures

Not applicable.

7.7 Additional Information [O]

Not applicable.

8 Non-Invasive Security

8.1 Mitigation Techniques [O]

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

8.2 Effectiveness [O]

Not applicable.

8.3 Additional Information [O]

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. The module does not perform persistent storage of SSPs	Dynamic

Table 16: 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	
AF_ALG type sockets (input)	Operator calling application (TOEPP)	Cryptographic module	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

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 APIs: appropriate zeroization functions: AES key: crypto_free_skcipher and crypto_free_aead; HMAC key: crypto_free_shash and crypto_free_ahash; DRBG entropy input string, DRBG seed, DRBG internal state: crypto_free_rng
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 18: 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, and computing MAC tags.	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			Symmetric encryption Symmetric decryption Message authentication Authenticated encryption Authenticated decryption
HMAC key	HMAC key used for: Message authentication, Authenticated encryption, Authenticated decryption.	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message authentication Authenticated encryption Authenticated decryption
DRBG entropy input string	DRBG entropy input used for: Random number generation. Compliant with IG D.L.	128-384 bits - 119-358 bits	Entropy Input - CSP			Random number generation
DRBG seed	DRBG seed derived from entropy input. Compliant with IG D.L.	CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRBG: 128, 256 bits - CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRB	Seed - CSP	Random number generation		Random number generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		G: 128, 256 bits				
DRBG internal state (V, Key)	DRBG internal state (V, Key) for HMAC and CTR DRBG. Compliant with IG D.L.	CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random number generation		Random number generation
DRBG internal state (V, C)	DRBG internal state (V, C) for Hash DRBG. Compliant with IG D.L.	Hash_DRBG: 128, 256 bits - Hash_DRBG: 128, 256 bits	Internal state - CSP	Random number generation		Random number generation

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters AF_ALG type sockets (input)	RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	
HMAC key	API input parameters AF_ALG type sockets (input)	RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	
DRBG entropy input string		RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	DRBG seed:Derives
DRBG seed		RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	DRBG entropy input string:Derived From DRBG internal state (V, Key):Derives DRBG internal

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					state (V, C):Derives
DRBG internal state (V, Key)		RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	DRBG seed:Derived From
DRBG internal state (V, C)		RAM:Plaintext	Until cipher handled is freed or module powered off	Wipe and Free memory block allocated Module Reset	DRBG seed:Derived From

Table 20: SSP Table 2

9.5 Transitions [O]

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

9.6 Additional Information [O]

Not applicable.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A4246)	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use.	Integrity test for static kernel binary.
HMAC-SHA2-256 (A4246)	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use.	Integrity test for fips_chk_hmac.

Table 21: 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.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A3599)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA-1 (A3603)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA-1 (A3604)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA-1 (A3605)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-224 (A3599)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-224 (A3603)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-224 (A3604)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-224 (A3605)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-256 (A3599)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-256 (A3603)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-256 (A3604)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-256 (A3605)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-384 (A3599)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-384 (A3603)	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
SHA2-384 (A3604)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-384 (A3605)	0-8184 bit messages	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-512 (A3599)	0-8184 bit message	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-512 (A3603)	0-8184 bit message	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-512 (A3604)	0-8184 bit message	KAT	CAST	Module becomes operational	Message digest	Module initialization
SHA2-512 (A3605)	0-8184 bit message	KAT	CAST	Module becomes operational	Message digest	Module initialization
AES-ECB (A3599)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-ECB (A3600)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-ECB (A3601)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-ECB (A3602)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CBC (A3599)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CBC (A3600)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CBC (A3601)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CBC (A3602)	128, 192, 256 bit keys;	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	encrypt and decrypt					
AES-CTR (A3599)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CTR (A3600)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CTR (A3601)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CTR (A3602)	128, 192, 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-XTS Testing Revision 2.0 (A3599)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-XTS Testing Revision 2.0 (A3600)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-XTS Testing Revision 2.0 (A3601)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-XTS Testing Revision 2.0 (A3602)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Module initialization
AES-CMAC (A3599)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Message authentication	Module initialization
AES-CMAC (A3601)	128 and 256 bit keys; encrypt and decrypt	KAT	CAST	Module becomes operational	Message authentication	Module initialization
AES-CMAC (A3602)	128 and 256 bit keys;	KAT	CAST	Module becomes operational	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	encrypt and decrypt					
HMAC-SHA-1 (A3599)	SHA-1 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA-1 (A3603)	SHA-1 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA-1 (A3604)	SHA-1 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA-1 (A3605)	SHA-1 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-224 (A3599)	SHA2-224 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-224 (A3603)	SHA2-224 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-224 (A3604)	SHA2-224 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-224 (A3605)	SHA2-224 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-256 (A3599)	SHA2-256 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-256 (A3603)	SHA2-256 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-256 (A3604)	SHA2-256 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-256 (A3605)	SHA2-256 with 32-64 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-384 (A3599)	SHA2-384 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-384 (A3603)	SHA2-384 with 32-	KAT	CAST	Module becomes operational	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	1048 bit keys					
HMAC-SHA2-384 (A3604)	SHA2-384 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-384 (A3605)	SHA2-384 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-512 (A3599)	SHA2-512 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-512 (A3603)	SHA2-512 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-512 (A3604)	SHA2-512 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
HMAC-SHA2-512 (A3605)	SHA2-512 with 32-1048 bit keys	KAT	CAST	Module becomes operational	Message authentication	Module initialization
Counter DRBG (A3599)	128, 192, 256 bit keys With DF, With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Counter DRBG (A3600)	128, 192, 256 bit keys With DF, With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Counter DRBG (A3601)	128, 192, 256 bit keys With DF, With/without PR; Health	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	test per section 11.3 of SP 800-90Arev1					
Counter DRBG (A3602)	128, 192, 256 bit keys With DF, With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Hash DRBG (A3599)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Hash DRBG (A3600)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Hash DRBG (A3601)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Hash DRBG (A3603)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A3604)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
Hash DRBG (A3605)	SHA-1, SHA2-256, SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
HMAC DRBG (A3599)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
HMAC DRBG (A3600)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
HMAC DRBG (A3601)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	of SP 800-90Arev1					
HMAC DRBG (A3603)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
HMAC DRBG (A3604)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
HMAC DRBG (A3605)	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Module initialization
ENT (NP)	1024 samples	RCT	CAST	Module becomes operational and services are available for use.	Entropy source start-up test	Entropy source initialization
ENT (NP)	1024 samples	APT	CAST	Module becomes operational and services are	Entropy source start-up test	Entropy source initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				available for use.		
ENT (NP)	Cutoff C = 61	RCT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
ENT (NP)	Cutoff C = 355	APT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
HMAC-SHA2-256 (A4249)	SHA2-256 with 256 bit key	KAT	CAST	Module becomes operational and services are available for use.	Message authentication. Makes use of HMAC from bound OpenSSL module.	Module initialization. Before integrity test.
HMAC-SHA2-256 (A4248)	SHA2-256 with 256 bit key	KAT	CAST	Module becomes operational and services are available for use.	Message authentication. Makes use of HMAC from bound OpenSSL module.	Module initialization. Before integrity test.
HMAC-SHA2-256 (A4247)	SHA2-256 with 256 bit key	KAT	CAST	Module becomes operational and services are available for use.	Message authentication. Makes use of HMAC from bound OpenSSL module.	Module initialization. Before integrity test.
HMAC-SHA2-256 (A4246)	SHA2-256 with 256 bit key	KAT	CAST	Module becomes operational and services are available for use.	Message authentication. Makes use of HMAC from bound OpenSSL module.	Module initialization. Before integrity test.

Table 22: 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)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4246)	Message Authentication	SW/FW Integrity	On demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
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
AES-ECB (A3599)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A3600)	KAT	CAST	On Demand	Manually
AES-ECB (A3601)	KAT	CAST	On Demand	Manually
AES-ECB (A3602)	KAT	CAST	On Demand	Manually
AES-CBC (A3599)	KAT	CAST	On Demand	Manually
AES-CBC (A3600)	KAT	CAST	On Demand	Manually
AES-CBC (A3601)	KAT	CAST	On Demand	Manually
AES-CBC (A3602)	KAT	CAST	On Demand	Manually
AES-CTR (A3599)	KAT	CAST	On Demand	Manually
AES-CTR (A3600)	KAT	CAST	On Demand	Manually
AES-CTR (A3601)	KAT	CAST	On Demand	Manually
AES-CTR (A3602)	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A3599)	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A3600)	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A3601)	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A3602)	KAT	CAST	On Demand	Manually
AES-CMAC (A3599)	KAT	CAST	On Demand	Manually
AES-CMAC (A3601)	KAT	CAST	On Demand	Manually
AES-CMAC (A3602)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3599)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3603)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3604)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3605)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
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
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

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Hash DRBG (A3604)	KAT	CAST	On Demand	Manually
Hash DRBG (A3605)	KAT	CAST	On Demand	Manually
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
ENT (NP)	RCT	CAST	On demand	Manually
ENT (NP)	APT	CAST	On demand	Manually
ENT (NP)	RCT	CAST	On demand	Manually
ENT (NP)	APT	CAST	On demand	Manually
HMAC-SHA2-256 (A4249)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4248)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4247)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4246)	KAT	CAST	On demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	The Linux kernel immediately stops executing	Any self-test failure	Restart of the module	Kernel Panic

Table 25: 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. Further details on how to recover from error state can be found in the evo-backup-snapshot documentation[1].

[1] URL: https://www.juniper.net/documentation/us/en/software/junos/junos-install-upgrade-evo/topics/topic-map/evo-backup-snapshot.html

10.5 Operator Initiation of Self-Tests [O]

All self-tests, with the exception of the continuous health tests, can be invoked on demand by unloading and subsequently re-initializing the module.

10.6 Additional Information [O]

Not applicable.

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[2]. 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 [SP800-131Ar2].
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

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

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 Design and Rules [O]

Not applicable for this module.

11.5 Maintenance Requirements [O]

There are no maintenance requirements.

11.6 End of Life [O]

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.

11.7 Additional Information [O]

Not applicable.

12 Mitigation of Other Attacks

12.1 Attack List [O]

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

12.2 Mitigation Effectiveness [O]

Not applicable.

12.3 Guidance and Constraints [O]

Not applicable.

12.4 Additional Information [O]

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
ENT (NP)	Non-physical Entropy Source
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, 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 198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.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 800-38B	**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 800-38E	**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
SP 800-90B	**Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP 800-131Ar2	**Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf