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

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.1

Last update: 2026-07-16

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 3.0.8 of the Junos OS Evolved OpenSSL 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 a Security Level 1 software module. It has a one-to-one mapping to SP 800-140B starting with section B.2.1 named "General" that maps to section 1 in this document and ending with section B.2.12 named "Mitigation of other attacks" that maps to section 12 in this document.

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	1
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Junos OS Evolved OpenSSL Cryptographic Module (hereafter referred to as "the module") is defined as a software module in a multi-chip standalone embodiment. The module is a software library which provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module consists of one software component, the "FIPS provider" (i.e. fips.so), which implements the FIPS requirements and the cryptographic functionality provided to the operator.

The module also uses the Junos OS Evolved Kernel Cryptographic Module Version 2.0 as a bound module (also referred to as "the bound Kernel Crypto API module") for performing random number generation, relying on the DRBG implemented in that bound module. The bound module is validated to FIPS 140-3 under Cert #5399. 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.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the fips.so shared library and its configuration file /etc/ssl/fipsmodule.cnf as shown in Figure 1 below.

Tested Operational Environment's Physical Perimeter (TOEPP):

Figure 1 shows a block diagram that represents the design of the module when the module is operational and providing services to other user space applications. In this diagram, the physical perimeter of the operational environment (a general-purpose computer on which the module is installed) is indicated by a purple dashed line. The cryptographic boundary is represented by orange shaded blocks.

Arrows show the flow of data between components. The labeled arrows indicate data flowing through the module's logical interfaces.

Components in white are only included in the diagram for informational purposes. They are not included in the cryptographic boundary (and therefore not part of the module's validation). For example, the kernel is responsible for managing system calls issued by the module itself, as well as other applications using the module for cryptographic services.

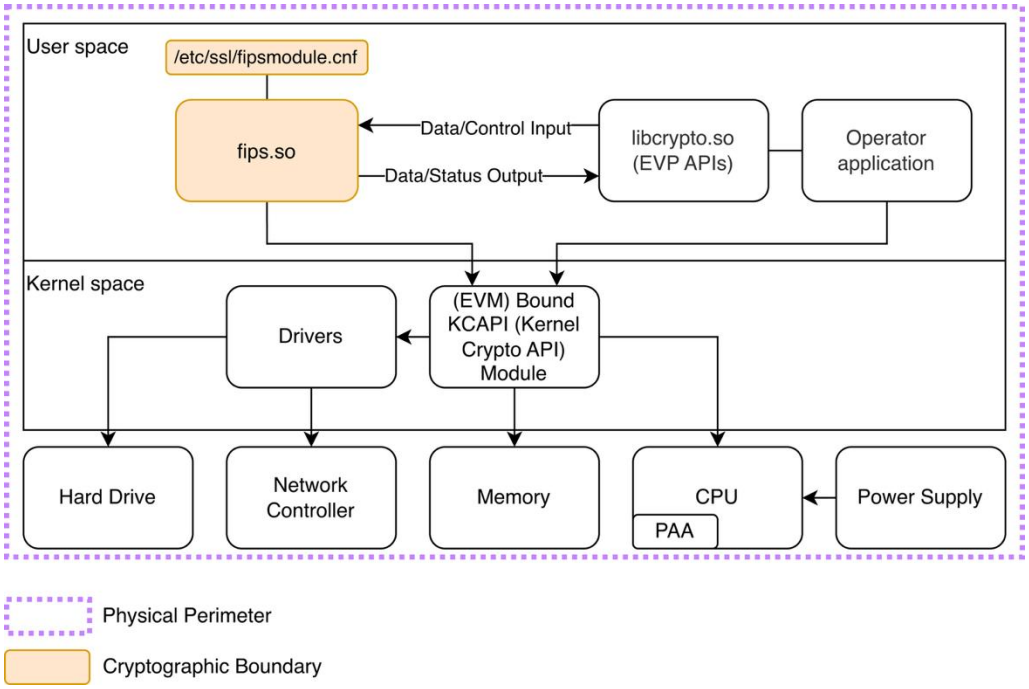

Figure 1: Block Diagram

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
fips.so and /etc/ssl/fipsmodule.cnf on Juniper Networks® Packet Transport Router Model PTX10001-36MR	3.0.8	N/A	HMAC-SHA2-256

Package or File Name	Software/ Firmware Version	Features	Integrity Test
with Intel® Xeon® D-2163IT			

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	3.0.8
Junos OS Evolved version 22.4	Juniper Networks® Packet Transport Router Model PTX10001-36MR	Intel® Xeon® D-2163IT	No	N/A	3.0.8

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

2.3 Excluded Components

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

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	Service API returns zero
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Service API returns non-zero value

Table 4: Modes List and Description

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on startup, the module automatically transitions to the approved mode. No operator intervention is required for this transition. The module continues to operate in the approved mode of operation by default and can only transition into the non-approved mode of operation by calling one of the non-approved services listed in the Non-Approved Services table in Section 4.4.

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	A4229, A4230, A4231	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A4229, A4230, A4231	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A
AES-CBC-CS2	A4229, A4230, A4231	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A
AES-CBC-CS3	A4229, A4230, A4231	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-65536 Increment 8	SP 800-38A
AES-CCM	A4229, A4230, A4231	Key Length - 128, 192, 256 Tag Length - 112, 128, 64, 80, 96 IV Length - IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length - Payload Length: 256 AAD Length - AAD Length: 0, 256, 65536	SP 800-38C
AES-CFB1	A4229, A4230, A4231	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4229, A4230, A4231	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4229, A4230, A4231	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A4229, A4230, A4231	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	A4229, A4230, A4231	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	A4229, A4230, A4231, A4232,	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
	A4233, A4234, A4235		
AES-GCM	A4237, A4238, A4239, A4240, A4241, A4242, A4243, A4244, A4245	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96, IV Length: 96, 128 Payload Length - Payload Length: 128, 256, 120, 248 AAD Length - AAD Length: 128, 256, 120, 0, AAD Length: 64, 96	SP 800-38D
AES-GMAC	A4237, A4238, A4239, A4240, A4241, A4242, A4243, A4244, A4245	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 128, 256, 120, 0	SP 800-38D
AES-KW	A4229, A4230, A4231	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F
AES-KWP	A4229, A4230, A4231	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 8-4096 Increment 8	SP 800-38F
AES-OFB	A4229, A4230, A4231	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4229, A4230, A4231	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
ECDSA KeyGen (FIPS186-4)	A4225	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A4246, A4247, A4248, A4249	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4225	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyVer (FIPS186-4)	A4246, A4247, A4248, A4249	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4225	Component - No Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4226	Component - No Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4236	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4246, A4247, A4248, A4249	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4225	Component - No Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4226	Component - No Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4236	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4246, A4247, A4248, A4249	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
HMAC-SHA-1	A4246, A4247, A4248, A4249	MAC - MAC: 160 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4246, A4247, A4248, A4249	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-256	A4246, A4247, A4248, A4249	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4246, A4247, A4248, A4249	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4246, A4247, A4248, A4249	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4246, A4247, A4248, A4249	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4246, A4247, A4248, A4249	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A4236	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A4236	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4236	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4236	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4227	Domain Parameter Generation Methods - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A4246, A4247, A4248, A4249	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A4251	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A4228	Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
KDA OneStep SP800-56Cr2	A4224	Auxiliary Function Methods - Auxiliary Function Name - SHA-1 Fixed Info Pattern - uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A4236	KDF Type - DER Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 zz Length - zz Length: 112-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-256 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.42 (CVL)	A4246, A4247, A4248, A4249	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 zz Length - zz Length: 112-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-256 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4246, A4247, A4248, A4249	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Field Size - 224, 571 Shared Info Length - Shared Info Length: 0-1024 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-135 Rev. 1
KDF SP800-108	A4250	KDF Mode - Counter, Feedback MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 Supported Lengths - Supported Lengths: 112-4096 Increment 8 Fixed Data Order - Before Fixed Data Counter Length - 32 Supports Empty IV - Yes Requires Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
KDF SSH (CVL)	A4232, A4233, A4234, A4235	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
KMAC-128	A4236	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - Yes Supports eXtendable-Output Functions - No, Yes	SP 800-185
KMAC-256	A4236	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - Yes Supports eXtendable-Output Functions - No, Yes	SP 800-185
PBKDF	A4236	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-132
PBKDF	A4246, A4247, A4248, A4249	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-4)	A4246, A4247, A4248, A4249	Key Generation Mode - B.3.6 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Info Generated By Server - No Public Exponent Mode - Random Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A4246, A4247, A4248, A4249	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-224	FIPS 186-4
RSA SigVer (FIPS186-4)	A4246, A4247, A4248, A4249	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA-1 Public Exponent Mode - Random	FIPS 186-4
Safe Primes Key Generation	A4251	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A4251	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4246, A4247, A4248, A4249	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A4236	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4236	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4236	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A4236	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A4236	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4236	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A4246, A4247, A4248, A4249	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Key Block Length - Key Block Length: 1024	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A4228	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 5: Approved Algorithms -

[EVM]

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	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		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	

Table 6: Approved Algorithms - [EVM]

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Asymmetric RSA Key Sizes:2048-15360 bits Safe Primes Fields:MODP-2048, ffdhe2048, MODP-3072, ffdhe3072, MODP-4096, ffdhe4096, MODP-6144, ffdhe6144, MODP-8192, ffdhe8192 ECC Curves:B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	N/A	SP 800-133 Rev. 2 Section 4 Example 1 with V=0
RSA Signature Generation with SHA-3	PKCS#1v1.5:SHA3-224, SHA3-256, SHA3-384, SHA3-512, 2048-16384 bits with 112-256 bits of security strength PSS:SHA3-224, SHA3-256, SHA3-384, SHA3-512, 2048-16384 bits with 112-256 bits of security strength IG C.C Compliance:Vendor affirmation claimed under IG C.C Resolution 2c.	N/A	FIPS 186-4
RSA Signature Verification with SHA-3	PKCS#1v1.5:SHA3-224, SHA3-256, SHA3-384, SHA3-512, 1024-16384 bits with 80-256 bits of security strength PSS:SHA3-224, SHA3-256, SHA3-384, SHA3-512, 1024-16384 bits with 80-256 bits of security strength IG C.C Compliance:Vendor affirmation claimed under IG C.C Resolution 2c.	N/A	FIPS 186-4

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES GCM (external IV)	Symmetric Encryption

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric Encryption	BC-UnAuth	Symmetric Encryption		AES-CBC: (A4229, A4230, A4231) AES-CBC-CS1: (A4229, A4230, A4231) AES-CBC-CS2: (A4229, A4230, A4231) AES-CBC-CS3: (A4229, A4230, A4231) AES-CFB1: (A4229, A4230, A4231) AES-CFB128: (A4229, A4230, A4231) AES-CFB8: (A4229, A4230, A4231) AES-CTR: (A4229, A4230, A4231) AES-ECB: (A4229, A4230, A4231, A4232, A4233, A4234, A4235) AES-OFB: (A4229, A4230, A4231) AES-XTS Testing Revision 2.0: (A4229, A4230, A4231)
Symmetric Decryption	BC-UnAuth	Symmetric Decryption		AES-CBC: (A4229, A4230, A4231) AES-CBC-CS1: (A4229, A4230, A4231) AES-CBC-CS2: (A4229, A4230, A4231) AES-CBC-CS3:

Name	Type	Description	Properties	Algorithms
				(A4229, A4230, A4231) AES-CFB1: (A4229, A4230, A4231) AES-CFB128: (A4229, A4230, A4231) AES-CFB8: (A4229, A4230, A4231) AES-CTR: (A4229, A4230, A4231) AES-ECB: (A4229, A4230, A4231, A4232, A4233, A4234, A4235) AES-OFB: (A4229, A4230, A4231) AES-XTS Testing Revision 2.0: (A4229, A4230, A4231)
Authenticated Symmetric Encryption	BC-Auth	Authenticated Symmetric Encryption		AES-CCM: (A4229, A4230, A4231) AES-GCM: (A4237, A4238, A4239, A4240, A4241, A4242, A4243, A4244, A4245)
Authenticated Symmetric Decryption	BC-Auth	Authenticated Symmetric Decryption		AES-CCM: (A4229, A4230, A4231) AES-GCM: (A4237, A4238, A4239, A4240, A4241, A4242, A4243, A4244, A4245)
Key Wrapping	BC-Auth	Key Wrapping as a service		AES-KW: (A4229, A4230, A4231) AES-KWP: (A4229, A4230, A4231)
Key Unwrapping	BC-Auth	Key Unwrapping as a service		AES-KW: (A4229, A4230, A4231) AES-KWP: (A4229, A4230, A4231)

Name	Type	Description	Properties	Algorithms
Message Digest	SHA XOF	Message Digest		SHA3-224: (A4236) SHA3-256: (A4236) SHA3-384: (A4236) SHA3-512: (A4236) SHAKE-128: (A4236) SHAKE-256: (A4236) SHA-1: (A4246, A4247, A4248, A4249) SHA2-224: (A4246, A4247, A4248, A4249) SHA2-256: (A4246, A4247, A4248, A4249) SHA2-384: (A4246, A4247, A4248, A4249) SHA2-512: (A4246, A4247, A4248, A4249) SHA2-512/224: (A4246, A4247, A4248, A4249) SHA2-512/256: (A4246, A4247, A4248, A4249)
MAC	MAC	Message Authentication Code		AES-CMAC: (A4229, A4230, A4231) HMAC-SHA3-224: (A4236) HMAC-SHA3-256: (A4236) HMAC-SHA3-384: (A4236) HMAC-SHA3-512: (A4236) KMAC-128: (A4236) KMAC-256: (A4236)

Name	Type	Description	Properties	Algorithms
				AES-GMAC: (A4237, A4238, A4239, A4240, A4241, A4242, A4243, A4244, A4245) HMAC-SHA-1: (A4246, A4247, A4248, A4249) HMAC-SHA2-224: (A4246, A4247, A4248, A4249) HMAC-SHA2-256: (A4246, A4247, A4248, A4249) HMAC-SHA2-384: (A4246, A4247, A4248, A4249) HMAC-SHA2-512: (A4246, A4247, A4248, A4249) HMAC-SHA2-512/224: (A4246, A4247, A4248, A4249) HMAC-SHA2-512/256: (A4246, A4247, A4248, A4249)
Random Number Generation	DRBG	[EVM] Random Number Generation	Hash:SHA2-512 Prediction resistaanace:No	HMAC DRBG: (A3599, A3600, A3601, A3603, A3604, A3605)
Key Pair Generation	AsymKeyPair-KeyGen	Key Pair Generation	RSA modulus size:2048-15360 bits Compliance:RSA modulus sizes other than 2048, 3072 and 4096 bits are not CAVP tested but approved per IG C.F	ECDSA KeyGen (FIPS186-4): (A4225, A4246, A4247, A4248, A4249) RSA KeyGen (FIPS186-4): (A4246, A4247, A4248, A4249) Safe Primes Key

Name	Type	Description	Properties	Algorithms
				Generation: (A4251)
Key Pair Verification	AsymKeyPair-KeyVer	Key Pair Verification		ECDSA KeyVer (FIPS186-4): (A4225, A4246, A4247, A4248, A4249) Safe Primes Key Verification: (A4251)
Signature Generation	DigSig-SigGen	Signature Generation	RSA modulus size:2048-16384 bits Compliance:RSA modulus sizes other than 2048, 3072 and 4096 bits are not CAVP tested but approved per IG C.F	ECDSA SigGen (FIPS186-4): (A4225, A4226, A4236, A4246, A4247, A4248, A4249) RSA SigGen (FIPS186-4): (A4246, A4247, A4248, A4249)
Signature Verification	DigSig-SigVer	Signature Verification	RSA modulus size:1024-16384 bits Compliance:RSA modulus sizes other than 1024, 2048, 3072 and 4096 bits are not CAVP tested but approved per IG C.F	ECDSA SigVer (FIPS186-4): (A4225, A4226, A4236, A4246, A4247, A4248, A4249) RSA SigVer (FIPS186-4): (A4246, A4247, A4248, A4249)
DH Shared Secret Computation	KAS-SSC	DH Shared Secret Computation		KAS-FFC-SSC Sp800-56Ar3: (A4251)
ECDH Shared Secret Computation	KAS-SSC	ECDH Shared Secret Computation		KAS-ECC-SSC Sp800-56Ar3: (A4227, A4246, A4247, A4248, A4249)
KDA OneStep Key Derivation	KAS-56CKDF	KDA OneStep Key Derivation		KDA OneStep SP800-56Cr2: (A4224)
HKDF Key Derivation	KAS-56CKDF	HKDF Key Derivation		KDA HKDF Sp800-56Cr1: (A4228)

Name	Type	Description	Properties	Algorithms
TLS 1.3 KDF Key Derivation	KAS-135KDF	TLS 1.3 KDF Key Derivation		TLS v1.3 KDF: (A4228)
SSH KDF Key Derivation	KAS-135KDF	SSH KDF Key Derivation		KDF SSH: (A4232, A4233, A4234, A4235)
ANS 9.42 KDF Key Derivation	KAS-135KDF	ANS 9.42 KDF Key Derivation		KDF ANS 9.42: (A4236, A4246, A4247, A4248, A4249)
Password-based Key Derivation	PBKDF	Password-based Key Derivation		PBKDF: (A4236, A4246, A4247, A4248, A4249)
ANS 9.63 KDF Key Derivation	KAS-135KDF	ANS 9.63 KDF Key Derivation		KDF ANS 9.63: (A4246, A4247, A4248, A4249)
TLS 1.2 KDF Key Derivation	KAS-135KDF	TLS 1.2 KDF Key Derivation		TLS v1.2 KDF RFC7627: (A4246, A4247, A4248, A4249)
KBKDF Key Derivation	KBKDF	KBKDF Key Derivation		KDF SP800-108: (A4250)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES-GCM

[EVM] When no IV is explicitly provided, the AES GCM IV generation is performed in compliance with Scenario 2 of IG C.H (Random IV). The AES-GCM IV is generated randomly internal to the module using the approved DRBG provided by the bound kernel module "Junos OS Evolved Kernel Cryptographic Module". The DRBG seeds itself from the entropy source of the kernel. The GCM IV is 96 bits in length.

2.7.2 AES-XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in SP 800-38E. The length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks, that is, 16MB of data. To meet the requirement in FIPS 140-3 IG C.I, the module implements a check to ensure that the two AES keys used in XTS-AES algorithm are not identical.

The two AES keys shall be generated and/or established independently according to the rules for component symmetric keys from SP 800-133 Rev. 2, Section 6.3.

2.7.3 PBKDF2

The module provides password-based key derivation (PBKDF2), compliant with [SP 800-132]. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance to SP 800-132 and FIPS 140-3 IG D.N, the following requirements shall be met:

- Derived keys shall only be used in storage applications. The MK shall not be used for other purposes. The module accepts a minimum length of 12 bits for the MK or DPK.
- Password and passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 8 characters. This will result in a password strength equal to 10^8. Combined with the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.
- [EVM] A portion of the salt, with a length of at least 128 bits (this is verified by the module to determine the service is approved), shall be generated randomly using the SP 800-90A Rev. 1 DRBG provided by the module.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum iteration count allowed by the module is 1000.

2.7.4 Diffie-Hellman and EC Diffie-Hellman

The module offers DH and ECDH shared secret computation services compliant to [SP 800-56A Rev. 3] and meeting IG D.F scenario 2 path (1). In order to meet the required assurances listed in Section 5.6 of [SP 800-56A Rev. 3], the module shall be used together with an application that implements the "TLS protocol" and the following steps shall be performed.

1. The entity using the module, must use the module's "Key pair generation" service for generating DH/ECDH ephemeral keys. This meets the assurances required by key pair owner defined in the section 5.6.2.1 of [SP 800-56A Rev. 3].
2. As part of the module's shared secret computation (SSC) service, the module internally performs the public key validation on the peer's public key passed in as input to the SSC function. This meets the public key validity assurance required by the sections 5.6.2.2.1/5.6.2.2.2 of [SP 800-56A Rev. 3].
3. The module does not support static keys therefore the "assurance of peer's possession of private key" is not applicable.

2.7.5 Key Wrapping

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

2.7.6 Key Agreement

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

2.7.7 RSA Key Length

The module supports RSA Key Generation and Signature Generation with any even modulus between 2048 and 16384 bits, and RSA Signature Verification with any even modulus between 1024 and 16384 bits. Moduli lengths other than 2048, 3072, and 4096 bits cannot be tested by CAVP but are approved for use as per IG C.F.

2.8 RBG and Entropy

[EVM] The module does not implement any random bit generator. Instead, the module obtains random bits from the Random Number Generation (RNG) service provided by the bound kernel module "Junos OS Evolved Kernel Cryptographic Module".

The bound kernel module uses the Kernel CPU Time Jitter RNG (validated under ESV certificate E50) as an entropy source to seed the DRBG. The entropy source is SP 800-90B compliant and is implemented within the bound kernel module. It resides within the TOEPP of the module and complies with IG 9.3.A Scenario 1b.

The DRBG implemented in the bound kernel module 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) from the entropy source. The 239 bits of entropy used for reseeding are less than the highest SSP strength generated by the module of 256 bits.

The module generates SSPs (e.g., keys) whose strengths are modified by available entropy.

2.9 Key Generation

The module provides the following key generation methods:

- Safe primes key pair generation: compliant with SP 800-56A Rev. 3. The method described in Section 5.6.1.1.4 of SP 800-56A Rev. 3 ("Testing Candidates") is used.
- RSA key pair generation: compliant with FIPS 186-4. The method described in Appendix B.3.6 of FIPS 186-4 ("Probable Primes with Conditions Based on Auxiliary Probable Primes") is used. Keys other than 2048, 3072 and 4096 bits are not CAVP tested but approved per IG C.F.
- ECC (ECDH and ECDSA) key pair generation: compliant with FIPS 186-4. The method described in Appendix B.4.2 of FIPS 186-4 ("Rejection Sampling") is used.

To obtain the random values used in asymmetric key pair generation, the module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133 Rev. 2 section 4 without the use of V (in accordance with additional comment 2 of IG D.H).

Additionally, the module implements the following key derivation methods:

- KBKDF: compliant with SP 800-108 Rev. 1. This implementation can be used to generate secret keys from a pre-existing key-derivation-key.
- KDA OneStep, HKDF: compliant with SP 800-56C Rev. 2. These implementations shall only be used to generate secret keys in the context of an SP 800-56A Rev. 3 key agreement scheme.
- ANSI X9.42 KDF, ANSI X9.63 KDF: compliant with SP 800-135 Rev. 1. These implementations shall only be used to generate secret keys in the context of an ANSI X9.42-2001 resp. ANSI X9.63-2001 key agreement scheme.
- SSH KDF, TLS 1.2 KDF, TLS 1.3 KDF: compliant with SP 800-135 Rev. 1. These implementations shall only be used to generate secret keys in the context of the SSH, TLS 1.2, or TLS 1.3 protocols, respectively.
- PBKDF2: compliant with option 1a of SP 800-132. This implementation shall only be used to derive keys for use in storage applications.

Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

2.10 Key Establishment

The module provides Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) shared secret computation compliant with SP 800-56A Rev. 3, in accordance with Scenario 2 (1) of FIPS 140-3 IG D.F.

For Diffie-Hellman, the module supports the following safe prime groups:

- For use in the IKE protocol (RFC 3526):
  - MODP-2048
  - MODP-3072
  - MODP-4096
  - MODP-6144
  - MODP-8192
- For use in the TLS protocol (RFC 7919):
  - ffdhe2048
  - ffdhe3072
  - ffdhe4096
  - ffdhe6144
  - ffdhe8192

For Elliptic Curve Diffie-Hellman, the module supports the NIST-defined B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, and P-521 curves.

According to FIPS 140-3 IG D.B, the key sizes of DH and ECDH shared secret computation provide 112-200 and 112-256 bits of security strength respectively in approved mode of operation.

The module also offers authenticated encryption and decryption as a service using AES-KW and AES-KWP. These algorithms can be used to wrap SSPs with a security strength of 128, 192, or 256 bits, depending on the wrapping key size.

2.11 Industry Protocols

The module implements the SSH key derivation function for use in the SSH protocol (RFC 4253 and RFC 6668).

The module implements the TLS 1.2 and TLS 1.3 key derivation functions for use in the TLS protocol.

The module implements the safe primes mentioned in Section 2.10 of this security policy.
No other parts of the SSH, TLS, or IKE protocols, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters, kernel I/O - network or files on file system, TLS protocol input messages
N/A	Data Output	API output parameters, kernel I/O - network or files on file system, TLS protocol output messages
N/A	Control Input	API function calls, API input parameters for control
N/A	Status Output	API return codes, error messages

Table 10: Ports and Interfaces

The logical interfaces are the APIs through which applications can request services from the module. These logical interfaces are logically separated from each other by the API design. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement authentication.

4.2 Roles

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

Table 11: Roles

The Crypto Officer role is implicitly and always assumed by the operator of the module. The module does not support multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Digest	Compute a message digest	EVP_Digest*() functions will return 0	Message	Message Digest	Message Digest	Crypto Officer
XOF	Compute the output of an XOF	EVP_Digest*() functions will return 0	Message, output length	XOF output of desired length	Message Digest	Crypto Officer
Symmetric Encryption	Encrypt a plaintext	EVP_Encrypt*() functions will return 0	Plaintext, AES key, IV	Ciphertext	Symmetric Encryption	Crypto Officer - AES key: W,E
Authenticated Symmetric Encryption	Encrypt & authenticate a plaintext	AES-CCM: EVP_Encrypt*() functions will return 0; AES-GCM: ERR_peek_last_error() function returns something different from 0x1C80012C	Plaintext, AES key, IV	Ciphertext, MAC tag	Authenticated Symmetric Encryption	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric Decryption	Decrypt a ciphertext	EVP_Decrypt*() functions will return 0	Ciphertext, AES key, IV	Plaintext	Symmetric Decryption Authenticated Symmetric Decryption	Crypto Officer - AES key: W,E
Authenticated Symmetric Decryption	Authenticate & decrypt a plaintext	EVP_Decrypt*() functions will return 0	Ciphertext, MAC tag, AES key, IV	Plaintext or Fail	Authenticated Symmetric Decryption	Crypto Officer - AES key: W,E
Key Wrapping	Perform AES-based key wrapping	EVP_Encrypt*() functions will return 0	Key to be wrapped, AES key wrapping key	Wrapped key	Key Wrapping	Crypto Officer - AES key: W,E
Key Unwrapping	Perform AES-based key unwrapping	EVP_Decrypt*() functions will return 0	Key to be unwrapped, AES key wrapping key	Unwrapped key	Key Unwrapping	Crypto Officer - AES key: W,E
Message Authentication Code	Compute a MAC tag	EVP_MAC*() functions will return 0	Message, MAC key (AES key, KMAC key, or HMAC key)	MAC tag	MAC	Crypto Officer - AES key: W,E - HMAC key: W,E - KMAC key: W,E
KBKDF Key Derivation	Derive a key from a key-derivation key	EVP_KDF*() functions will return 0	Key-derivation key	KBKDF derived key	KBKDF Key Derivation	Crypto Officer - Key-derivation key: W,E - KBKDF derived key: G,R
KDA OneStep Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	KDA OneStep derived key	KDA OneStep Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - KDA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						OneStep derived key: G,R
HKDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	Hkdf derived key	HKDF Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - HKDF derived key: G,R
ANS X9.42 KDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	ANS X9.42 KDF derived key	ANS 9.42 KDF Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - ANS X9.42 derived key: G,R
ANS X9.63 KDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	ANS X9.63 KDF derived key	ANS 9.63 KDF Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - ANS X9.63 derived key: G,R
SSH KDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	SSH KDF derived key	SSH KDF Key Derivation None	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - SSH KDF derived key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TLS 1.2 KDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	TLS 1.2 KDF derived key	TLS 1.2 KDF Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - TLS 1.2 KDF derived key: G,R
TLS 1.3 KDF Key Derivation	Derive a key from a shared secret	EVP_KDF*() functions will return 0	Shared secret	TLS 1.3 KDF derived key	TLS 1.3 KDF Key Derivation	Crypto Officer - DH shared secret: W,E - ECDH shared secret: W,E - TLS 1.3 KDF derived key: G,R
Password-based Key Derivation	Derive a key from a password	EVP_KDF*() functions will return 0	Password	PBKDF derived key	Password-based Key Derivation	Crypto Officer - Password: W,E - PBKDF2 derived key: G,R
DH Shared Secret Computation	Compute a shared secret	EVP_PKEY*() functions will return 0	DH private key, DH public key from peer	DH shared secret	DH Shared Secret Computation	Crypto Officer - DH private key: W,E - DH public key: W,E - DH shared secret: G,R
ECDH Shared Secret Computation	Compute a shared secret	EVP_PKEY*() functions will return 0	ECDH private key, ECDH public key from peer	ECDH shared secret	ECDH Shared Secret Computation	Crypto Officer - ECDH private key: W,E - ECDH public key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ECDH shared secret: G,R
Signature Generation	Generate a signature	EVP_DigestSign*() functions will return 0	Message, Private key (RSA private key or ECDSA private key)	Signature	Signature Generation	Crypto Officer - RSA private key: W,E - ECDSA private key: W,E
Signature Verification	Verify a signature	EVP_DigestVerify*() functions will return 0	Message, Public key (RSA public key or ECDSA public key), Signature	Pass or Fail	Signature Verification	Crypto Officer - RSA public key: W,E - ECDSA public key: W,E
Key Pair Generation	Generate a key pair	EVP_PKEY*() will return 0	Domain (group, curve, or bitlength)	Key pair (DH, EC, or RSA)	Key Pair Generation	Crypto Officer - DH private key: G,R - DH public key: G,R - ECDH private key: G,R - ECDH public key: G,R - ECDSA private key: G,R - ECDSA public key: G,R - RSA private key: G,R - RSA public key: G,R - Intermediate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key Generation Value: G,E,Z
Key Pair Verification	Verify a key pair	EVP_PKEY*() will return 0	Domain (group or curve), Key pair (DH or EC)	Pass or Fail	Key Pair Verification	Crypto Officer - DH private key: W - DH public key: W - ECDH private key: W - ECDH public key: W - ECDSA private key: W - ECDSA public key: W
[EVM] Random Number Generation	Generate random bytes	RAND_bytes*() will return 0	Output length	Random bytes	Random Number Generation	Crypto Officer - [EVM] DRBG entropy input string: W,E,Z - [EVM] HMAC_DRBG seed: G,E,Z - [EVM] HMAC_DRBG internal state: G,W,E
Show Version	Return the name and version information	N/A	N/A	Module Version	None	Crypto Officer
Show Status	Return the module status	N/A	N/A	Module Status	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Self-test	Perform the CASTs and integrity test	N/A	N/A	Pass or Fail	Random Number Generation KDA OneStep Key Derivation Key Pair Generation Key Pair Verification Signature Generation Signature Verification ECDH Shared Secret Computation HKDF Key Derivation TLS 1.3 KDF Key Derivation Symmetric Encryption Symmetric Decryption Authenticated Symmetric Encryption Authenticated Symmetric Decryption MAC Key Wrapping Key Unwrapping SSH KDF Key Derivation ANS 9.42 KDF Key	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Derivation Password-based Key Derivation Message Digest ANS 9.63 KDF Key Derivation TLS 1.2 KDF Key Derivation KBKDF Key Derivation DH Shared Secret Computation	
Zeroization	Zeroize CPSs	N/A	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - KMAC key: Z - Key-derivation key: Z - DH shared secret: Z - ECDH shared secret: Z - Password: Z - KBKDF derived key: Z - KDA OneStep derived key: Z - HKDF derived key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- ANS X9.42 derived key: Z - ANS X9.63 derived key: Z - SSH KDF derived key: Z - TLS 1.2 KDF derived key: Z - TLS 1.3 KDF derived key: Z - PBKDF2 derived key: Z - DH private key: Z - DH public key: Z - ECDH private key: Z - ECDH public key: Z - ECDSA private key: Z - ECDSA public key: Z - RSA private key: Z - RSA public key: Z

Table 12: Approved Services

To interact with the module, a calling application must use the EVP API layer provided by OpenSSL. This layer will delegate the request to the FIPS provider, which will in turn perform the requested service.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM (external IV)	Symmetric Encryption	AES GCM (external IV)	CO

Table 13: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support the loading of external software/firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC SHA2-256 value calculated at run time on the fips.so binary with the HMAC SHA2-256 value stored in the /etc/ssl/fipsmodule.cnf file that was computed during installation of the module. The HMAC key used for this integrity test is hardcoded into the utility which performs the integrity check.

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 test may be invoked on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. This will perform (among others) the software integrity test.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

All SSPs contained within the module are protected by process isolation and memory separation mechanisms enforced by the Junos OS Evolved operating system. The module thus has sole control over its SSPs.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11. If properly installed, 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.

Instrumentation tools like the ptrace system call, gdb and strace utilities, userspace live patching, as well as other tracing mechanisms offered by the Linux environment such as ftrace or systemtap, shall not be used in the operational environment. The use of any of these tools implies that the cryptographic module is running in a non-tested 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 14: Storage Areas

SSPs are provided to the module by the calling application and are destroyed when released by the appropriate API function calls. The module does not perform persistent storage of SSPs.

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	
API output parameters	Cryptographic module	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	

Table 15: SSP Input-Output Methods

The module does not support the input or output of cryptographically protected SSPs.

The module only supports SSP input and output to and from a calling application running on the same operational environment. This corresponds to manual distribution, electronic entry/output ("CM Software to/from App via TOEPP Path") per FIPS 140-3 IG 9.5.A Table 1.

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	By calling the cipher related zeroization API functions: EVP_CIPHER_free for AES keys not used in MACs; EVP_MAC_CTX_free for AES keys used for GMAC or CMAC, HMAC keys, and KMAC keys; EVP_KDF_CTX_free for derived keys, shared secrets, and

Zeroization Method	Description	Rationale	Operator Initiation
		indicates that the zeroization procedure succeeded.	passwords; EVP_PKEY_free for asymmetric keys
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 within nanoseconds when power is removed.	By unloading and reloading the module

Table 16: SSP Zeroization Methods

The application that uses the module is responsible for calling the API functions which perform zeroization of SSPs stored in the module. The module provides key destruction functions which overwrite the memory containing SSPs and their contexts.

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, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			Symmetric Encryption Symmetric Decryption Authenticated Symmetric Encryption Authenticated Symmetric Decryption MAC Key Wrapping Key Unwrapping
HMAC key	HMAC key	112-256 bits - 112-256 bits	Symmetric key - CSP			MAC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KMAC key	KMAC key	128, 192, 256 bits - 129, 192, 256 bits	Symmetric key - CSP			MAC
Key-derivation key	Key-derivation key	112-256 bits - 112-256 bits	Key-derivation key - CSP			KBKDF Key Derivation
DH shared secret	DH shared secret	MODP-2048, ffdhe2048, MODP-3072, ffdhe3072, MODP-4096, ffdhe4096, MODP-6144, ffdhe6144, MODP-8192, ffdhe8192 - 112, 128, 149, 172, 200 bits	Shared Secret - CSP		DH Shared Secret Computation	KDA OneStep Key Derivation HKDF Key Derivation TLS 1.3 KDF Key Derivation SSH KDF Key Derivation ANS 9.42 KDF Key Derivation ANS 9.63 KDF Key Derivation TLS 1.2 KDF Key Derivation
ECDH shared secret	ECDH shared secret used	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Shared Secret - CSP		ECDH Shared Secret Computation	KDA OneStep Key Derivation HKDF Key Derivation TLS 1.3 KDF Key Derivation SSH KDF Key Derivation ANS 9.42 KDF Key Derivation ANS 9.63 KDF Key Derivation TLS 1.2 KDF

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Key Derivation
Password	Password	8-128 bytes - N/A	Password - CSP			Password-based Key Derivation
KBKDF derived key	KBKDF derived key	112-4096 bits - 112-256 bits	Symmetric key - CSP	KBKDF Key Derivation		
KDA OneStep derived key	KDA OneStep derived key	2048 bits - 112-256 bits	Symmetric key - CSP	KDA OneStep Key Derivation		
HKDF derived key	HKDF derived key	2048 bits - 112-256 bits	Symmetric key - CSP	HKDF Key Derivation		
ANS X9.42 derived key	ANS X9.42 derived key	112-4096 bits - 112-256 bits	Symmetric key - CSP	ANS 9.42 KDF Key Derivation		
ANS X9.63 derived key	ANS X9.63 derived key	128-4096 bits - 112-256 bits	Symmetric key - CSP	ANS 9.63 KDF Key Derivation		
SSH KDF derived key	SSH KDF derived key	112-4096 - 112-256 bits	Symmetric key - CSP	SSH KDF Key Derivation		
TLS 1.2 KDF derived key	TLS 1.2 KDF derived key	112-4096 bits - 112-256 bits	Symmetric key - CSP	TLS 1.2 KDF Key Derivation		
TLS 1.3 KDF derived key	TLS 1.3 KDF derived key	112-4096 bits - 112-256 bits	Symmetric key - CSP	TLS 1.3 KDF Key Derivation		
PBKDF2 derived key	PBKDF2 derived key	128-4096 bits - 112-256 bits	Symmetric key - CSP	Password-based Key Derivation		
DH private key	DH private key	MODP-2048, ffdhe2048, MODP-3072, ffdhe3072, MODP-4096, ffdhe4096, MODP-	Private key - CSP	Key Pair Generation		Key Pair Verification DH Shared Secret Computation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		6144, ffdhe6144, MODP-8192, ffdhe8192 - 112, 128, 149, 172, 200 bits				
DH public key	DH public key	MODP-2048, ffdhe2048, MODP-3072, ffdhe3072, MODP-4096, ffdhe4096, MODP-6144, ffdhe6144, MODP-8192, ffdhe8192 - 112, 128, 149, 172, 200 bits	Public key - PSP	Key Pair Generation		Key Pair Verification DH Shared Secret Computation
ECDH private key	ECDH private key	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation		ECDH Shared Secret Computation
ECDH public key	ECDH public key	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation		ECDH Shared Secret Computation
RSA private key	RSA private key	2048-16384 bits - 112-256 bits	Private key - CSP	Key Pair Generation		Key Pair Verification Signature Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA public key	RSA public key	1024-16384 bits - 80-256 bits	Public key - PSP	Key Pair Generation		Key Pair Verification Signature Verification
ECDSA private key	ECDSA private key	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation		Key Pair Verification Signature Generation
ECDSA public key	ECDSA public key	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation		Key Pair Verification Signature Verification
Intermediate Key Generation Value	Intermediate key pair generation value generated during key generation and key derivation services (SP 800-133 Rev. 2 Section 4, 5.1, and 5.2)	112-8192 bits - 112-256 bits	Intermediate value - CSP	Key Pair Generation		Key Pair Generation
[EVM] DRBG entropy input string	Entropy input string for DRBG in bound module (IG D.L compliant)	256-384 bits - 238-358 bits	Entropy Input - CSP			Random Number Generation
[EVM] HMAC_DRBG seed	DRBG seed derived from entropy input in bound module (IG D.L compliant)	512 bits - 256 bits	Seed - CSP	Random Number Generation		Random Number Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
[EVM] HMAC_DRBG internal state	Internal state of DRBG in bound module (IG D.L compliant)	1024 bits - 256 bits	Seed - CSP	Random Number Generation		Random Number Generation

Table 17: 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	
KMAC key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
Key-derivation key	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	KBKDF derived key:Derives
DH shared secret	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	KDA OneStep derived key:Derived From HKDF derived key:Derived From SSH KDF derived key:Derived From TLS 1.2 KDF derived key:Derived From TLS 1.3 KDF derived key:Derived From DH private

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					key:Established from DH public key:Established from
ECDH shared secret	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	KDA OneStep derived key:Derived From HKDF derived key:Derived From SSH KDF derived key:Derived From TLS 1.2 KDF derived key:Derived From TLS 1.3 KDF derived key:Derived From ECDH private key:Established from ECDH public key:Established from
Password	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	PBKDF2 derived key:Derives
KBKDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Key-derivation key:Derived From
KDA OneStep derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
HKDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
ANS X9.42 derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ANS X9.63 derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
SSH KDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
TLS 1.2 KDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
TLS 1.3 KDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Derived From ECDH shared secret:Derived From
PBKDF2 derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Password:Derived From
DH private key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Establishes DH public key:Paired With Intermediate Key Generation Value:Derived From
DH public key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DH shared secret:Establishes DH private key:Paired With Intermediate Key Generation Value:Derived From
ECDH private key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block	ECDH shared secret:Establishes ECDH public key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				allocated Module Reset	Intermediate Key Generation Value:Derived From
ECDH public key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	ECDH shared secret:Establishes ECDH private key:Paired With Intermediate Key Generation Value:Derived From
RSA private key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA public key:Paired With Intermediate Key Generation Value:Derived From
RSA public key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA private key:Paired With Intermediate Key Generation Value:Derived From
ECDSA private key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	ECDSA public key:Paired With Intermediate Key Generation Value:Derived From
ECDSA public key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	ECDSA private key:Paired With Intermediate Key Generation Value:Derived From
Intermediate Key Generation Value		RAM:Plaintext	From service invocation to service completion	Automatic Module Reset	DH private key:Derives DH public key:Derives ECDH private key:Derives ECDH public key:Derives RSA private key:Derives RSA public key:Derives ECDSA private

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					key:Derives ECDSA public key:Derives
[EVM] DRBG entropy input string		RAM:Plaintext	From service invocation to service completion	Automatic Module Reset	[EVM] HMAC_DRBG seed:Derives
[EVM] HMAC_DRBG seed		RAM:Plaintext	From service invocation to service completion	Automatic Module Reset	[EVM] DRBG entropy input string:Derived From [EVM] HMAC_DRBG internal state:Derives
[EVM] HMAC_DRBG internal state		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Automatic Module Reset	[EVM] HMAC_DRBG seed:Derived From

Table 18: SSP Table 2

10 Self-Tests

The module performs the pre-operational self-test and CASTs automatically when the module is loaded into memory. The pre-operational integrity test is only executed after all cryptographic algorithm self-tests (CASTs) have executed successfully.

While the module is executing the pre-operational test and the CASTs, the module services are not available, and data output is inhibited. The module is not available for use by the calling application until the pre-operational self-test and the CASTs are completed successfully. After the pre-operational test and the CASTs succeed, the module becomes operational. If any of the pre-operational test or any of the CASTs fail an error message is returned, and the module transitions to the error state.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A4246)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4247)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4248)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4249)	SHA2-256	MAC tag verification	SW/FW Integrity	Module becomes operational	N/A

Table 19: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A4246)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4247)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4248)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A4249)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4246)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4247)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4248)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4249)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A4236)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-512 (A4236)		KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
AES-GCM (A4237) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4238) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4239) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4240) - Encryption	256-bit key, 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-GCM (A4241) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4242) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4243) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4244) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4245) - Encryption	256-bit key, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4237) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4238) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4239) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4240) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4241) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A4242) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4243) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4244) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4245) - Decryption	256-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4229) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4230) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4231) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4232) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4233) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4234) - Decryption	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4235) - Decryption	128-bit key, 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
HMAC-SHA-1 (A4246)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A4247)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A4248)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A4249)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A4246)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A4247)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A4248)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A4249)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4246)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4247)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A4248)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4249)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A4246)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A4247)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A4248)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A4249)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A4246)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A4247)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A4248)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A4249)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
KDF SP800-108 (A4250)	HMAC-SHA2-256, counter mode	KAT	CAST	Module becomes operational	Key based key derivation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
KDA OneStep SP800-56Cr2 (A4224)	SHA2-224	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4228)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4236)	SHA-1, AES-KW with 128-bit key	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4246)	SHA-1, AES-KW with 128-bit key	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4247)	SHA-1, AES-KW with 128-bit key	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4248)	SHA-1, AES-KW with 128-bit key	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4249)	SHA-1, AES-KW with 128-bit key	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4246)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4247)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4248)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF ANS 9.63 (A4249)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF SSH (A4232)	SHA-1	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A4233)	SHA-1	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A4234)	SHA-1	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A4235)	SHA-1	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4246)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4247)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4248)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4249)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.3 KDF (A4228)	SHA2-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.3 KDF key derivation	Test runs at power-on before the integrity test
PBKDF (A4236)	SHA2-256, 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	iterations, 288-bit salt					before the integrity test
PBKDF (A4246)	SHA2-256, 4096 iterations, 288-bit salt	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4247)	SHA2-256, 4096 iterations, 288-bit salt	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4248)	SHA2-256, 4096 iterations, 288-bit salt	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4249)	SHA2-256, 4096 iterations, 288-bit salt	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4246)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4247)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4248)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4249)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4246)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4247)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-4) (A4248)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4249)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4225)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4226)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4236)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4246)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4247)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4248)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4249)	P-224, B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4225)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4226)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
ECDSA SigVer (FIPS186-4) (A4236)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4246)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4247)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4248)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4249)	P-224. B-233 with SHA-224	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-4) (A4225)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4246)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4247)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4248)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4249)	SHA-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA KeyGen (FIPS186-4) (A4246)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4247)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4248)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4249)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
KAS-FFC-SSC Sp800-56Ar3 (A4251) - ffdhe	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A4251) - MODP	MODP-2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4227)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4246)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4247)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4248)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4249)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
Safe Primes Key	N/A	PCT	PCT	Successful key pair generation	Key pair generation	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Generation (A4251)						
HMAC DRBG (A3599)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A3600)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A3601)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A3603)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A3604)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A3605)	HMAC-SHA2-512 without prediction resistance	KAT	CAST	Module becomes operational	[EVM] SP 800-90A Rev. 1 (Instantiate, reseed, generate) health test	Test runs at power-on before the integrity test

Table 20: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4246)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A4247)	MAC tag verification	SW/FW Integrity	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4248)	MAC tag verification	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-256 (A4249)	MAC tag verification	SW/FW Integrity	On Demand	Manually

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA-1 (A4246)	KAT	CAST	On Demand	Manually
SHA-1 (A4247)	KAT	CAST	On Demand	Manually
SHA-1 (A4248)	KAT	CAST	On Demand	Manually
SHA-1 (A4249)	KAT	CAST	On Demand	Manually
SHA2-512 (A4246)	KAT	CAST	On Demand	Manually
SHA2-512 (A4247)	KAT	CAST	On Demand	Manually
SHA2-512 (A4248)	KAT	CAST	On Demand	Manually
SHA2-512 (A4249)	KAT	CAST	On Demand	Manually
SHA3-256 (A4236)	KAT	CAST	On Demand	Manually
SHA3-512 (A4236)	KAT	CAST	On Demand	Manually
AES-GCM (A4237) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4238) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4239) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4240) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4241) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4242) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4243) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4244) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4245) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4237) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4238) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4239) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4240) - Decryption	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A4241) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4242) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4243) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4244) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4245) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4229) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4230) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4231) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4232) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4233) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4234) - Decryption	KAT	CAST	On Demand	Manually
AES-ECB (A4235) - Decryption	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4246)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4247)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4248)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A4249)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A4246)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A4247)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A4248)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A4249)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4246)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4247)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4248)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4249)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4246)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4247)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4248)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A4249)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A4246)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A4247)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A4248)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A4249)	KAT	CAST	On Demand	Manually
KDF SP800-108 (A4250)	KAT	CAST	On Demand	Manually
KDA OneStep SP800-56Cr2 (A4224)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4228)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4236)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4246)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4247)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4248)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4249)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4246)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4247)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4248)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDF ANS 9.63 (A4249)	KAT	CAST	On Demand	Manually
KDF SSH (A4232)	KAT	CAST	On Demand	Manually
KDF SSH (A4233)	KAT	CAST	On Demand	Manually
KDF SSH (A4234)	KAT	CAST	On Demand	Manually
KDF SSH (A4235)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4246)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4247)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4248)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4249)	KAT	CAST	On Demand	Manually
TLS v1.3 KDF (A4228)	KAT	CAST	On Demand	Manually
PBKDF (A4236)	KAT	CAST	On Demand	Manually
PBKDF (A4246)	KAT	CAST	On Demand	Manually
PBKDF (A4247)	KAT	CAST	On Demand	Manually
PBKDF (A4248)	KAT	CAST	On Demand	Manually
PBKDF (A4249)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4246)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4247)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4248)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4249)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4246)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4247)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4248)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A4249)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4225)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4226)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4236)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4246)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4247)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4248)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4249)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4225)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4226)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4236)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4246)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4247)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4248)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-4) (A4249)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4225)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4246)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4247)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4248)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4249)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4246)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4247)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4248)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4249)	PCT	PCT	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A4251) - ffdhe	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A4251) - MODP	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4227)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4246)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KAS-ECC-SSC Sp800-56Ar3 (A4247)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4248)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4249)	KAT	CAST	On Demand	Manually
Safe Primes Key Generation (A4251)	PCT	PCT	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

Table 22: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	General purpose error	Software integrity test failure CAST failure PCT failure	Restart module	Module will not load; module is aborted (for PCT failure)

Table 23: Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process. Consequently, the data output interface is inhibited, and the module no longer accepts inputs or service requests as the application is not running.

10.5 Operator Initiation of Self-Tests

The operator can initiate the pre-operational self-tests and the cryptographic algorithms self-tests by unloading and subsequently re-initializing the module.

The operator can also initiate the pre-operational self-test by calling the OSSL_PROVIDER_self_test function.

Pair-wise consistency tests can be invoked on demand by requesting the key generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The binaries of the module are contained in the base Junos Evolved installation image.

The operator is responsible to verify the correct installation of module which is already pre-installed on the image file (junos-evo-install-ptx-fixed-x86-64-22.4R2.11-S1-EVO.iso).

The following steps are required:

- run the following command:

  openssl fipsinstall -module /usr/lib64/ossl-modules/fips.so -in /etc/ssl/openssl-fips.cnf -provider_name fips -verify

  Which should output the following:

  VERIFY PASSED

- run the following command to request the “Show module name and version” service and check the name and the version of the OpenSSL:

  openssl list -provider

  Which should output the following:

  Providers:
  base
  name: OpenSSL Base Provider
  version: 3.0.8
  status: active
  fips
  name: Junos OS Evolved OpenSSL Cryptographic Module
  version: 3.0.8
  status: active

11.2 Administrator Guidance

The Crypto Officer shall follow section 11.1 of this Security Policy to verify that the module is installed correctly. The Crypto Officer shall follow this Security Policy to operate the module as a FIPS 140-3 validated module.

11.3 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory.

12 Mitigation of Other Attacks

12.1 Attack List

Certain cryptographic subroutines and algorithms are vulnerable to timing analysis. The module mitigates this vulnerability by using constant-time implementations. This includes, but is not limited to:

- Big number operations: computing GCDs, modular inversion, multiplication, division, and modular exponentiation (using Montgomery multiplication).
- Elliptic curve point arithmetic: addition and multiplication (using the Montgomery ladder).
- Vector-based AES implementations.

In addition, RSA, ECDSA, ECDH, and DH employ blinding techniques to further impede timing and power analysis. No configuration is needed to enable these countermeasures.

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	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** August 2020 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 800-38C	**Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality** May 2004 https://doi.org/10.6028/NIST.SP.800-38C
SP 800-38D	**Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 https://doi.org/10.6028/NIST.SP.800-38D
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
SP 800-38F	**Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://doi.org/10.6028/NIST.SP.800-38F
FIPS 180-4	**Secure Hash Standard (SHS)** August 2015 https://doi.org/10.6028/NIST.FIPS.180-4
FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://doi.org/10.6028/NIST.FIPS.202

SP 800-185	**SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash** December 2016 https://doi.org/10.6028/NIST.SP.800-185
FIPS 198-1	**The Keyed-Hash Message Authentication Code (HMAC)** July 2008 https://doi.org/10.6028/NIST.FIPS.198-1
FIPS 186-4	**Digital Signature Standard (DSS)** July 2013 https://doi.org/10.6028/NIST.FIPS.186-4
SP 800-132	**Recommendation for Password-Based Key Derivation Part 1: Storage Applications** December 2010 https://doi.org/10.6028/NIST.SP.800-132
SP 800-108 Rev. 1	**Recommendation for Key Derivation Using Pseudorandom Functions** August 2022 https://doi.org/10.6028/NIST.SP.800-108r1-upd1
SP 800-56A Rev. 3	**Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 https://doi.org/10.6028/NIST.SP.800-56Ar3
SP 800-56C Rev. 2	**Recommendation for Key-Derivation Methods in Key-Establishment Schemes** August 2020 https://doi.org/10.6028/NIST.SP.800-56Cr2
SP 800-135 Rev. 1	**Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 https://doi.org/10.6028/NIST.SP.800-135r1
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 800-133 Rev. 2	**Recommendation for Cryptographic Key Generation** June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** Feburary 2003 https://www.ietf.org/rfc/rfc3447.txt

RFC 3526 **More Modular Exponential (MODP) Diffie-Hellman groups for Intenet Key Exchange (IKE)**
May 2003
https://www.ietf.org/rfc/rfc3526.txt

RFC 4253 **The Secure Shell (SSH) Transport Layer Protocol**
January 2006
https://www.ietf.org/rfc/rfc4253.txt

RFC 6668 **SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol**
July 2012
https://www.ietf.org/rfc/rfc6668.txt

RFC 7919 **Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)**
August 2016
https://www.ietf.org/rfc/rfc7919.txt

ANSI X9.42 **Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography**
2013
https://webstore.ansi.org/Standards/ASCX9/ANSIX9422003R2013

ANSI X9.63 **Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography**
2017
https://webstore.ansi.org/Standards/ASCX9/ANSIX9632011R2017