HPE Juniper Networking

HPE Juniper Networking QFX5120-48YM

FIPS 140-3 Non-Proprietary Security Policy

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

Introduction
Federal Information Processing Standards Publication 140-3 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation

Program (CMVP) run the FIPS 140-3 program. The NVLAP accredits independent testing labs to perform FIPS 140-3 testing; the CMVP validates modules meeting FIPS 140-3 validation. Validated is the term given to a module that is documented and tested against the FIPS 140-3 criteria.

More information is available on the CMVP website at:
https://csrc.nist.gov/projects/cryptographic-module-validation-program.

About this Document
This is a non-proprietary Cryptographic Module Security Policy for the HPE Juniper Networking QFX5120-48YM, Release 23.4R2 provides an overview of the product and a high-level description of how it meets the overall Level 1, security requirements of FIPS 140-3.

Disclaimer
The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. HPE Juniper Networking shall have no liability for any error or damages of any kind resulting from the use of this document.

Notices
This document may be freely reproduced and distributed in its entirety without modification.

This document describes the cryptographic module security policy for the HPE Juniper Networking QFX5120-48YM cryptographic module (also referred to as the “module” hereafter) with firmware version Junos OS 23.4R2. The module has a multi-chip standalone embodiment. It contains specification of the security rules, under which the cryptographic module operates, including the security rules derived from the requirements of the FIPS 140-3 standard.

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	3
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	1
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

1.3 Additional Information

The module claims an overall Security Level of 1 with all individual sections at a Security Level 1 with the exceptions of Roles, Services and Authentication (claimed at Security Level 3). The module does not implement any non-invasive security mitigations or mitigations of other attacks and thus the requirements per these sections are inapplicable.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The cryptographic module provides for an encrypted connection, using SSH, between the management station and the module. The cryptographic module also provides for an encrypted connection, using MACsec, between itself and a peer.

Module Type: Hardware

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic module's operational environment is a limited operational environment. The cryptographic boundary of the hardware module is the entirety of the module/chassis. This includes the Routing Engine (RE). No components have been excluded from the cryptographic boundary of the module.

Tested Operational Environment's Physical Perimeter:

The cryptographic boundary is the entirety of the module chassis.

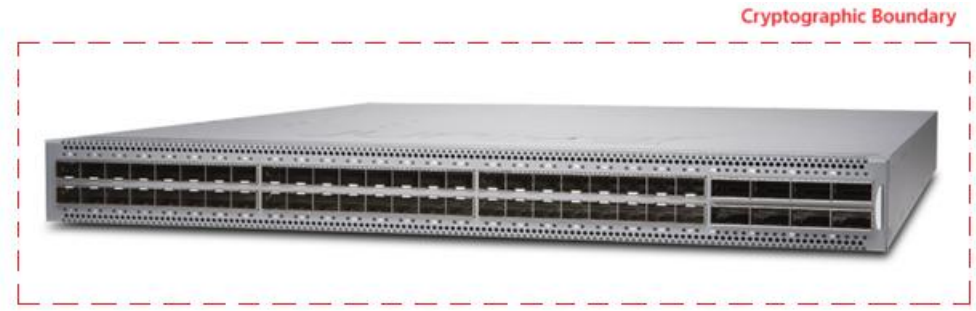

Figure 1: Front view of QFX5120-48YM

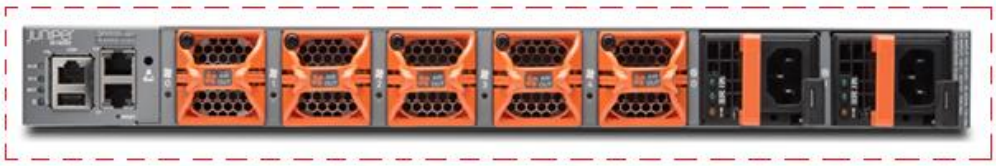

Figure 2: Rear view for QFX5120-48YM

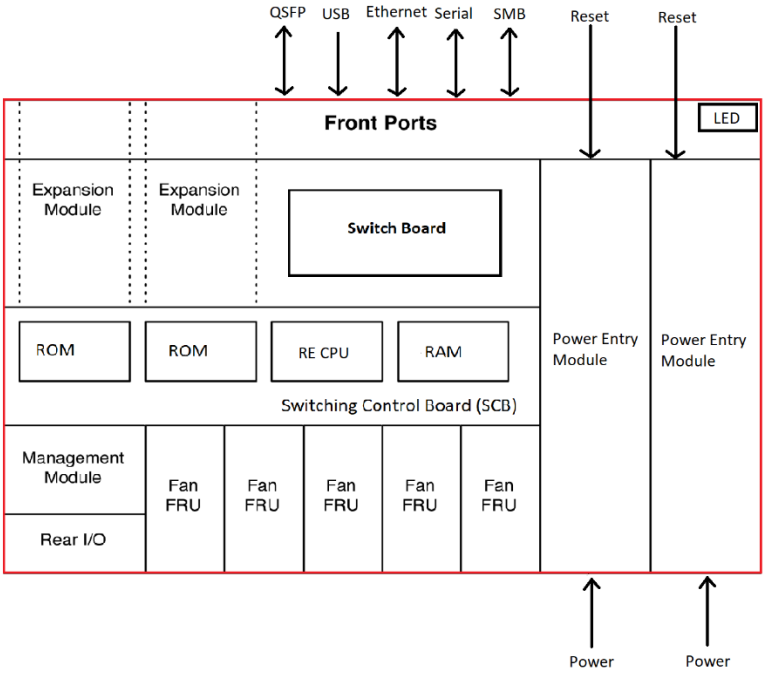

Figure 3: Block Diagram for QFX5120-48YM

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
QFX5120-48YM	QFX5120-48YM	Junos OS 23.4R2	Intel Xeon D-1627 (Hewitt Lake-DE)	N/A

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

No components have been excluded from the cryptographic boundary of the module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	• The operator can verify that the cryptographic module is in the Approved mode by observing the console prompt and running the “show version” command; • When operating in the Approved mode, the prompt will read “<operator>:fips#” (e.g. root:fips#); • The “show version” command will allow the Crypto Officer to verify that the validated firmware version is running on the module; • The Crypto Officer can also use the “show system fips chassis level” command (returns “level 1”) to determine if the module is operating in the Approved mode; • The Approved mode is entered when the module is configured for it and successfully passes all self-tests (both pre-operational and conditional cryptographic algorithm self-tests (CASTs))	Approved	global indicator (string 'fips' included in the command prompt)
Non-Approved mode	• The cryptographic module supports a non-Approved mode of operation; • When operated in the non-Approved mode of operation, the module supports non-Approved algorithms as well as the algorithms supported in the Approved mode of operation	Non-Approved	global indicator (implicit indicator based on exclusion of string 'fips' from the command prompt)

Table 3: Modes List and Description

The hardware version in Table 2, with Junos OS 23.4R2 installed, contains one Approved mode of operation and a non-Approved mode of operation. The Junos OS 23.4R2 firmware image must first be installed on the module. When operated in the non-Approved mode of operation, the module supports non-Approved algorithms as well as the algorithms supported in the Approved mode of operation.

Mode Change Instructions and Status:

The module is in the non-approved mode upon installation of the module firmware, and the Crypto Officer can place the module into the Approved mode of operation by following the instructions/commands provided below:

[edit]
root# request system zeroise

[edit]

root# set system fips chassis level 1

[edit]

root# show system fips chassis level
level 1;

To switch from the Approved mode of operation back to the non-Approved mode of operation the module has to be zeroised again using the following command:

[edit]
root# request system zeroize

Degraded Mode Description:

The module does not support a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6551	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A7286	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A6551	Direction - Generation, Verification Key Length - 128, 256 MAC Length - MAC Length: 8, 64, 120 Message Length - Message Length: 0, 32768, 524288	SP 800-38B
AES-CTR	A7286	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-GCM	AES 4545	Direction - Decrypt, Encrypt Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 120, 128 AAD Length - AAD Length: 120, 128	SP 800-38D
AES-KW	A6551	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128	SP 800-38F

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 128, 192, 256, 320, 4096	
AES-XPN	AES 4545	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 120, 128 AAD Length - AAD Length: 120, 128 Tag Length - 104, 112, 120, 128, 64, 96 Salt Generation - External	SP 800-38D
ECDSA KeyGen (FIPS186-5)	A7286	Curve - P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A7286	Curve - P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7286	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A7286	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC DRBG	A6620	Prediction Resistance - Yes Supports Reseed - Yes Mode - SHA2-512 Entropy Input - Entropy Input: 256 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 8 Additional Input - Additional Input: 8-256 Increment 8 Returned Bits - 1024	SP 800-90A Rev. 1
HMAC-SHA-1	A7286	MAC - MAC: 160 Key Length - Key Length: 160	FIPS 198-1
HMAC-SHA2-256	A6620	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A7286	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-512	A7286	MAC - MAC: 512 Key Length - Key Length: 512	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7286	Domain Parameter Generation Methods - P-256, P-384, P-521 Hash Function Z - SHA2-256, SHA2-384, SHA2-512 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
KAS-FFC-SSC Sp800-56Ar3	A7286	Domain Parameter Generation Methods - FC, MODP-2048 Hash Function Z - SHA2-256 Scheme - dhEphem - KAS Role - initiator	SP 800-56A Rev. 3
KDF SP800-108	A6551	KDF Mode - Counter MAC Mode - CMAC-AES128, CMAC-AES256 Supported Lengths - Supported Lengths: 128, 256 Fixed Data Order - Before Fixed Data Counter Length - 8 Supports Empty IV - No Custom Key In Length - 0	SP 800-108 Rev. 1
KDF SSH (CVL)	A7286	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A7286	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Fixed Public Exponent - 010001 Info Generated By Server - No Private Key Format - standard Public Exponent Mode - fixed	FIPS 186-5
RSA SigGen (FIPS186-5)	A7286	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A7286	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5 Fixed Public Exponent - 010001 Public Exponent Mode - fixed	FIPS 186-5
Safe Primes Key Generation	A7286	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
Safe Primes Key Verification	A7286	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
SHA-1	A7286	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A6620	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A7286	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6550	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512	A6620	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A7286	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

The following protocol is supported by the module in the Approved mode:

SSHv2 (EC Diffie-Hellman P-256, P-384, P-521; Diffie-Hellman MODP2048; RSA 2048, 3072 4096 bits; ECDSA P-256, P-384, P-521; AES CBC 128, 192, 256 bits; AES CTR 128, 192, 256 bits, HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-512)

MACsec (MACsec Key Agreement (MKA); AES GCM, XPN 128 and 256 bits)

The SSH protocol allows independent selection of key exchange, authentication, cipher and integrity algorithms. Please note that there are algorithms, modes, and key/moduli sizes that have been CAVP-tested but are not used by any approved service of the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the table above are used by an approved service of the module.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG - Section 4	Key Type:Symmetric and Asymmetric	N/A	NIST SP800-133r2 Section 4: Symmetric key generation and Asymmetric seed generation using an unmodified output from an Approved DRBG (example 1); The module supports the following per NIST SP 800-133r2: 1. Section 5.1: Key Pairs for Digital Signature Schemes 2. Section 5.2: Key Pairs for Key Establishment 3. Section 6.2.1: Derivation of symmetric keys

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

The module does not support any non-Approved algorithms in the Approved mode, i.e., it does not support Non-Approved Algorithms Allowed in the Approved Mode of Operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
SHA2-256 (Junos 23.4R2 - LibMD Implementation)	no security claimed	Used to store operator passwords in hashed form, per IG 2.4.A: Use of a non-approved cryptographic algorithm to "obfuscate" a CSP

Name	Caveat	Use and Function
SHA-1 (Junos 23.4R2 - Kernel Implementation)	no security claimed	Used for an extraneous check in the Kernel, per IG 2.4.A: Use of an approved, non-approved or proprietary algorithm for a purpose that is not security relevant

Table 6: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
RSA with key size less than 2048	SSH
ECDSA with ed25519 curve	SSH
EC Diffie-Hellman with ed25519 curve	SSH
ARCFOUR	SSH
Blowfish	SSH
CAST	SSH
DSA (SignGen, SigVer, non-compliant)	SSH
HMAC-MD5	SSH
HMAC-RIPEMD160	SSH
UMAC	SSH
TDES	SSH

Table 7: Non-Approved, Not Allowed Algorithms

In addition to the above non-Approved Algorithms Not Allowed in the Approved Mode of Operation, all Approved algorithms supported in the Approved mode of operation are also supported in the non-Approved mode.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
KAS1	CKG KAS-135KDF KAS-Full KAS-SSC	Key Agreement for SSHv2	IG: IG D.F Scenario 2, path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides between 128	KAS-ECC-SSC Sp800-56Ar3: (A7286) KDF SSH: (A7286) CKG - Section 4 : () Key Type: Symmetric and Asymmetric

Name	Type	Description	Properties	Algorithms
			and 256 bits of security strength	
KAS2	CKG KAS-135KDF KAS-Full KAS-SSC	Key Agreement for SSHv2	IG:IG D.F Scenario 2, path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL Caveat:Key establishment methodology provides 112 bits of security strength	KAS-FFC-SSC Sp800-56Ar3: (A7286) KDF SSH: (A7286) CKG - Section 4 : () Key Type: Symmetric and Asymmetric Safe Primes Key Generation: (A7286) Safe Primes Key Verification: (A7286)
KTS1	KTS-Wrap	Key Transport for SSHv2	Standard:SP 800-38F IG D.G:approved method from IG D.G Key confirmation:no Caveat:Key establishment methodology provides between 128 and 256 bits of security strength	AES-CBC: (A7286) AES-CTR: (A7286) HMAC-SHA-1: (A7286) HMAC-SHA2-256: (A7286) HMAC-SHA2-512: (A7286) SHA-1: (A7286) SHA2-256: (A7286) SHA2-512: (A7286)

Name	Type	Description	Properties	Algorithms
ECDSA SigVer	DigSig-SigVer	ECDSA Signature Verification used for firmware integrity	FIPS 186-5 :size: P-256, encryption strength: 128 bits	ECDSA SigVer (FIPS186-5): (A7286)
ECDSA SigVer2	DigSig-SigVer	ECDSA Signature Verification used for identity-based public key authentication	FIPS 186-5:size: P-256, P-384, P-521 curves, 128, 192 and 256 bits	ECDSA SigVer (FIPS186-5): (A7286)
DRBG	DRBG	Kernel DRBG providing random bits for SSP generation in the user/application space		HMAC DRBG: (A6620) HMAC-SHA2-256: (A6620) SHA2-256: (A6620)
Entropy Source	ENT-Cond	Non-Physical Entropy Source		SHA2-512: (A6620)
ECDSA KeyGen	AsymKeyPair-KeyGen	Generation of SSH host keys		ECDSA KeyGen (FIPS186-5): (A7286) CKG - Section 4 : () Key Type: Symmetric and Asymmetric
ECDSA KeyGen2	AsymKeyPair-KeyGen	SSP Agreement in the context of SSH		ECDSA KeyGen (FIPS186-5): (A7286) CKG - Section 4 : () Key Type: Symmetric and Asymmetric
ECDSA KeyVer	AsymKeyPair-KeyVer	Verification of keys generated		ECDSA KeyVer (FIPS186-5): (A7286)

Name	Type	Description	Properties	Algorithms
ECDSA SigGen	DigSig-SigGen	Signature Generation using ECDSA in the context of SSH		ECDSA SigGen (FIPS186-5): (A7286)
RSA KeyGen	AsymKeyPair-KeyGen	Generation of SSH host keys		RSA KeyGen (FIPS186-5): (A7286) CKG - Section 4 : () Key Type: Symmetric and Asymmetric
RSA SigGen	DigSig-SigGen	Signature Generation using RSA in the context of SSH		RSA SigGen (FIPS186-5): (A7286)
RSA SigVer	DigSig-SigVer	Signature Verification using RSA for public key authentication		RSA SigVer (FIPS186-5): (A7286)
Password Hash	SHA	Used to store passwords in hashed form		SHA2-512: (A6550)
MACsec Encryption/Decryption	BC-Auth	Encryption/Decryption of MACsec packets		AES-GCM: (AES 4545) AES-XPN: (AES 4545)
KTS2	KTS-Wrap	Key Transport for MACsec	Standard:SP800-38D IG D.G:approved method from IG D.G Key confirmation:no Caveat :Key establishment methodology provides between 128 and 256 bits of security strength	AES-KW: (A6551)

Name	Type	Description	Properties	Algorithms
MACsec Key Derivation	KBKDF MAC	NIST SP 800-108 KDF used in the context of MAcsec to derive SSPs		KDF SP800-108: (A6551) AES-CMAC: (A6551) AES-CBC: (A6551)
CASTs on boot	BC-Auth BC-UnAuth DigSig-SigGen DigSig-SigVer DRBG ENT-Cond KAS-135KDF KBKDF MAC SHA	List of algorithms for which Known Answer Tests (CASTs) have been implemented in the module and perform on each boot		AES-CBC: (A7286) HMAC-SHA-1: (A7286) HMAC-SHA2-256: (A7286, A6620) HMAC-SHA2-512: (A7286) KAS-ECC-SSC Sp800-56Ar3: (A7286) KAS-FFC-SSC Sp800-56Ar3: (A7286) KDF SSH: (A7286) ECDSA SigGen (FIPS186-5): (A7286) ECDSA SigVer (FIPS186-5): (A7286) RSA SigGen (FIPS186-5): (A7286) RSA SigVer (FIPS186-5): (A7286)

Name	Type	Description	Properties	Algorithms
				HMAC DRBG: (A6620) SHA2-512: (A6620, A6550) AES-GCM: (AES 4545) AES-KW: (A6551) KDF SP800-108: (A6551) AES-CMAC: (A6551)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

IG 2.4.B
Per the Resolution in the IG, the following component has been tested and documented as a CVL in the module's validation certificate:

4. KDF per 135r1: KDF SSH per CAVP Cert. #A7286
The Security Policy shall individually list the tested components shown in the module's CVL certificates that may be called during the operation of the module: The KDF SSH (CVL) has been listed separately in the Security Policy Section 2.5, Approved Algorithms Table.

Additional Comments
#4. SP 800-135rev1 and the TLS 1.3 KDFs are considered approved CVLs only when performed in the context of their respective protocols: The Security Policy contains the following statement in Section 2.11: No parts of the SSH and MACsec protocols, other than the KDF SSH and the NIST SP 800-108 KDF for MACsec, have been tested by the CAVP or CMVP. The module meets the requirements per FIPS 140-3 IG IG D.C #2. Per FIPS 140-3 IG 2.4.B, the KDF SSH is considered an approved CVL only when performed in the context of the SSHv2 protocol.

The Security Policy shall individually list the vendor affirmed components that may be called during the operation of the module: The only vendor affirmed algorithm, CKG, has been separately documented in the Vendor Affirmed algorithms table.

IG C.H
The GCM IV is constructed in the context of the IETF MACsec protocol. Scenario 1. c. per FIPS 140-3 IG C.H applies to the module.

MACsec protocol IV generation
The AES GCM IV construction is performed internal to the module in compliance with IEEE 802.1AEand its amendments. The IV length is 96 bits (per SP 800-38D). The module ensures the IV is constructed deterministically per Section 8.2 in SP 800-38D and the MACsec standard IEEE 802.1AE as a result of concatenating the fixed field (SCI) and invocation field (PN).

The following shall statements apply (the Security Policy contains the cited statements in Section 2.12 #13 a. to e.):
For the purposes of a FIPS 140-3 validation, the cryptographic functionality relevant for MACsec in each component shall be validated as a separate module. The requirement in Section 9.1 in SP 800-38D to contain the IV “generation unit” within a module boundary is satisfied by the composition of the Peer, Authenticator and optional Authentication Server modules. All modules – Peer, Authenticator, and, if applicable, Authentication Server, should be validated (or re-validated) after this Implementation Guidance takes effect, so that they all comply with the applicable requirements of this IG.

While all modules are validated separately by the CMVP, each module’s Security Policy shall tell what this module’s role is in the MACsec protocol, explain what the module does in support of the IV generation for the MACsec’s use of AES-GCM, and state that when supporting the MACsec protocol in the approved mode, the module should only be used together with the CMVP-validated modules providing the remaining functionalities:

The module can take on the role of Peer or Authenticator in reference to the MACsec protocol. The module shall only be used with other FIPS 140-3 validated modules when supporting the MACsec protocol in the role of a Peer/Authenticator for providing the remaining functionalities. The module shall satisfy one of the IV restoration conditions defined in Scenario 3 of this Implementation Guidance: If the module loses power and then it is restored, then a new key shall be established for use with the AES CMAC encryption/decryption processes.

The Peer and the Authenticator Modules Security Policies shall state that the link between the Peer and the Authenticator should be secured to prevent the possibility for an attacker introducing foreign equipment into the local area network: The link between the Peer and Authenticator, used in the MACsec communication shall be secure to prevent the possibility for an attacker introducing foreign equipment into the local area network.

In accordance with Additional Comment #8 in the IG, a static SAK is not configured.

IG D.L
Section 7.1 of SP 800-90Ar1 states: “[T]he entropy input and the seed shall be kept secret.” Therefore, the entropy input string and the seed shall be considered CSPs for all the DRBG mechanisms. As stated in Section 8.3 of SP 800-90A, some values of the working state are considered secret values of the internal state. Therefore, they shall be considered CSPs as well. These values are listed below:

HMAC_DRBG mechanism The values of V and Key are the “secret values” of the internal state.

As can be verified from the Security Policy Section 9.4 SSP Table, the HMAC_DRBG Key, V, Entropy Input and Seed have been deemed/defined CSPs.

IG D.M

The role of the SP 800-108 KDFs is to derive new keys from existing keying material. Therefore, all key derivation methods listed in SP 800-108 are approved for use in an approved mode if the Key Derivation Key as introduced in Section 5 of SP 800-108 has been generated, entered or established using a method approved or allowed for keys in an approved mode. SP 800-108 KDFs may not be used to generate asymmetric keys.

As can be verified from the Security Policy Section 9.4, SSP Table (for keys where the establishment method is “Derived using SP800-108 KDF”, the “Security Function” column shows that only symmetric keys are generated),the module only uses the NIST SP800-108 KDF to derive symmetric keys.

IG C.E and C.F
The module's RSA CAVP Cert. #A7286 meets the requirements of both IGs as approved moduli 2048, 3072 and 4096 have been tested (approved Key Generation, Signature Generation and Signature Verification). No untested moduli apply.

The minimum number of the Miller-Rabin tests used in primality testing are consistent with the bit sizes of p, q, p1, p2, q1 and q2 taken from Table B.1 (for the row corresponding to mod 3072 given that the largest modulus supported by the module is 4096 bits) of FIPS 186-5 (per Table C.2 entries in FIPS 186-5).

IG C.B
The following apply to and are met by the module (all approved hash algorithms have been CAVP tested): every approved hash algorithm implementation shall be CAVP tested and validated on all of the module’s operating environments. For higher level cryptographic algorithms that use these approved hash algorithms (e.g. RSA, ECDSA, KBKDF, HMAC, SP 800-135 KDFs, SP 800-56C KDFs, etc.), every implemented combination for which CAVP testing exists, shall, upon the expiration of a transitional period defined when such CAVP testing becomes available, be CAVP tested and validated on all of the module’s operating environments.

2.8 RBG and Entropy

Cert Number	Vendor Name
E215	HPE Juniper Networking

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Junos OS 23.4 Non-Physical Entropy Source	Non-Physical	Intel Xeon D-1627 (Hewitt Lake-DE)	512 bits	448 bits	SHA2-512 (CAVP Cert. #A6620)

Table 10: Entropy Sources

The module generates 256 bits of entropy which is sufficient for the generation of the SSPs (using the approved DRBGs of the module) with the maximum target security strength (256 bits) needed.

As can be verified from the Public Use Document and the ESV Cert. #E215, the module generates a minimum of 448 bits of overall entropy per 512-bit output sample (entropy input to the DRBG) for SSP generation.

2.9 Key Generation

The module implements a NIST SP 800-90Ar1 DRBG and supports the following sections per NIST SP 800-133r2 (CKG): Sections 4, 5.1, 5.2 and 6.2.1.

2.10 Key Establishment

Per IG D.F:

The module implements full KAS (KAS-ECC-SSC, KAS-FFC-SSC per NIST SP 800-56Ar3 and KDF SSH per NIST SP 800-135r1; IG D.F Scenario 2 (path 2 option 2, separate testing of the SSC and SP800-135r1 KDF). The KAS1 and KAS2 in the SFI Table have been documented in accordance with this requirement:

KAS1: KAS (KAS-ECC-SSC Cert. #A7286 and CVL Cert. #A7286; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

KAS2: KAS (KAS-FFC-SSC Cert. #A7286 and CVL Cert. #A7286; SSP establishment methodology provides 112 bits of encryption strength)

The Approved Algorithm list includes the tested components (KAS-ECC-SSC, KAS-FFC-SSC and KDF SSH) as individual entries.

Per IG D.G:
The module supports the IETF SSH and MACsec protocols and thus implements key transport in the context of the protocols (per the KTS1 and KTS2 entries in the SFI table of the Security Policy).

The module implements the following approved KTS using approved AES modes:

AES CBC and CTR (KTS1): KTS (AES Cert. #A7286 and HMAC Cert. #A7286; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

AES KW (KTS2): KTS (AES Cert. #A6551; SSP establishment methodology provides between 128 and 256 bits of encryption strength)

2.11 Industry Protocols

Per IG D.C: No parts of the SSH and MACsec protocols, other than the KDF SSH and the NIST SP 800-108 KDF for MACsec, have been tested by the CAVP or CMVP.

2.12 Additional Information

The module design corresponds to the security rules below. The term *shall* in this context specifically refers to a requirement for correct usage of the module in the Approved mode; all other statements indicate a security rule implemented by the module.

1. The module clears previous authentications on power cycle.
2. When the module has not been placed in a valid role, the operator does not have access to any cryptographic services.
3. Self-tests do not require any operator action.
4. Data output is inhibited during SSP generation, self-test execution, zeroisation, and error states.
5. Status information does not contain SSPs or sensitive data that if misused could lead to a compromise of the module.
6. There are no restrictions on which SSPs are zeroised by the zeroisation service.
7. The module does not support a maintenance interface or role.
8. The module does not output intermediate key values.
9. The module does not output plaintext CSPs.
10. The Crypto officer shall verify that the firmware image to be loaded on the module is a FIPS 140-3 validated image. If any non-validated firmware image is loaded the module will no longer be a validated module.
11. The Crypto Officer shall retain control of the module while zeroisation is in process.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Ethernet (Management Port)	Data Input Data Output Control Input Status Output	RJ-45 management port (labeled C1 and C0) (No. of ports: 2)
Serial	Data Input Data Output Control Input Status Output	RJ-45 Console Serial Port (No. of ports: 1)
USB	Data Input Control Input	USB port, load Junos Image (No. of ports: 1)
Power	Power	Power supplies, (No. of ports: 2)
Alarm LEDs	Status Output	Chassis status LEDs (labeled ALM, SYS, MST, and ID) (No. of LEDs: 4); Power supply LEDs (No. of LEDs: 2); Fan module LEDs (No. of LEDs: 5)

Physical Port	Logical Interface(s)	Data That Passes
Reset Button	Control Input	Reset (No. of ports: 1)
10M OUT and PPS OUT ports	Data Input Data Output	10M OUT and PPS OUT ports, (No. of ports: 2)

Table 11: Ports and Interfaces

The module does not support control output.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Username and password over the console and SSH	* The module enforces 10-character passwords (at minimum) chosen from the 96 human readable ASCII characters; The maximum password length is 20-characters; Thus, the probability of a successful random attempt is 1/(96^10), which is less than 1/1,000,000 (million); * The module enforces a timed access mechanism as follows: For the first two failed attempts (assuming 0 time to process), no timed access is enforced; Upon the third attempt, the module enforces a 5-second delay; Each failed attempt thereafter results in an additional 5-second delay above the previous (e.g., 4th failed attempt = 10-second delay, 5th failed attempt = 15-second delay, 6th failed attempt = 20-second delay, 7th failed attempt = 25-second delay); This leads to a maximum of 7 possible attempts in a one-minute period for each getty; The best approach for the attacker would be to disconnect after 4 failed attempts and wait for a new getty	SHA2-512 (A6550)	1/(96^10)	9/(96^10)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	to be spawned; This would allow the attacker to perform roughly 9.6 attempts per minute (576 attempts per hour/60 mins); this would be rounded down to 9 per minute, because there is no such thing as 0.6 attempts; The probability of a success with multiple consecutive attempts in a one-minute period is 9/(96^10), which is less than 1/100,000			
Username and ECDSA public key over SSH	* The module supports ECDSA (P-256, P-384, and P-521), which has a minimum equivalent computational resistance to attack of either 2^128, 2^192 or 2^256 depending on the curve; Thus, the probability of a successful random attempt is 1/(2^128), which is less than 1/1,000,000 (million) * Configurable SSH connection establishment rate limits the number of connection attempts, and thus failed authentication attempts in a one-minute period to a maximum of 15,000 attempts; The probability of a success with multiple consecutive attempts in a one-minute period is 15,000/(2^128), which is less than 1/100,000	ECDSA SigVer (FIPS186-5) (A7286)	1/(2^128)	15,000/(2^128)
Username and RSA public key over SSH	* The module supports RSA (2048, 3072, 4096 bits), which has a minimum equivalent computational resistance to attack of 2^112 (2048 bits); Thus, the probability of a successful random attempt is 1/ (2^112), which is less than 1/1,000,000 (million) * Configurable SSH connection establishment rate limits the number of connection attempts, and thus failed authentication attempts in a one-minute period to a maximum of 15,000 attempts; The probability of a success with multiple	RSA SigVer (FIPS186-5) (A7286)	1/ (2^112)	15,000/(2^112)

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
	consecutive attempts in a one-minute period is 15,000/(2^112), which is less than 1/100,000			

Table 12: Authentication Methods

The module enforces the separation of roles using identity-based operator authentication. The module implements two forms of identity-based authentication, username, and password over the console and SSH connections, as well as username and an ECDSA or RSA public key-based authentication over SSHv2.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Super-user	Identity	Crypto Officer (CO)	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Operator	Identity	User	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Read-only	Identity	User	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Root	Identity	Crypto Officer (CO)	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH
Unauthorised	Identity	User	Username and password over the console and SSH Username and ECDSA public key over SSH Username and RSA public key over SSH

Table 13: Roles

The module supports two roles: Crypto Officer (CO) and User. Root and Super-user correspond to the Crypto Officer role whereas Operator, Read-Only and Unauthorised operator types correspond to the User role. The module supports concurrent operators but does not support a maintenance role and/or bypass capability.

An operator assuming the Crypto Officer role configures and monitors the module via a console or SSH connection. As Root or Super-user, the Crypto Officer has permission to view and configure passwords and public keys within the module. The User role monitors the module via the console or SSH. The User role does not have the permission to modify the configuration.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure security (security relevant)	Security relevant configuration (SSH, authentication data)	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Commands (SSH configuration: set system services ssh root-login allow)	Traffic	DRBG Entropy Source Password Hash	Root - SSH Private Host Key: G - User Password: W,E - CO Password: W,E - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - SSH Public Host Key: G - User Authentication Public Keys: W - CO Authentication Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Keys: W Super-user - SSH Private Host Key: G - User Password: W,E - CO Password: W,E - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - SSH Public Host Key: G - User Authentication Public Keys: W - CO Authentication Public Keys: W
Configure (non-security relevant)	Non-security relevant configuration	Global Approved Mode indicator "fips" at the CLI combined with succes	Commands (miscellaneous commands e.g., for IP address configuration, routing protocols, etc.)	Traffic	Password Hash	Super-user - CO Password: E Root - CO Password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		sful completion of each service				
Show status	Query the module status	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Command (show system fips chassis level, requests system fips self-tests)	CLI output (show system fips chassis level: returns “level 1” to indicate that the module is operating in the Approved mode and no output/blank to indicate that it is operating in the non-Approved mode; request system fips self-tests: module continues to be operational upon successful execution/returns an error indicator and enters an error state in case of a failure)	Password Hash	Super-user - CO Password: E Root - CO Password: E Operator - User Password: E Read-only - User Password: E Unauthorised - User Password: E
Show status (LED)	LEDs on the module provide physical	LED(s) on the chassis turned on	N/A	LED	None	Super-user Operator Read-only Unauthorised Root

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	status output					Unauthenticated
Show module's versioning information	Query the module's versioning information	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Command (show version)	CLI output	Password Hash	Super-user - CO Password: E Operator - User Password: E Read-only - User Password: E Unauthorised - User Password: E Root - CO Password: E
Zeroise (Perform zeroisation)	Destroy all SSPs	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Command (request vmhost zeroise no-forwarding)	N/A	Password Hash	Super-user - SSH Private Host Key: Z - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - User Password: Z - CO Password: E,Z - HMAC_DRBG V value: Z - HMAC_DR

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						BG Key value: Z - HMAC_DRBG entropy input: Z - HMAC_DRBG seed: Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: Z - SSH Public Host Key: Z - User Authentication Public Keys: Z - CO Authentication Public Keys: Z - JuniperRootCA: Z - PackageCA: Z - SSH ECDH Public Key: Z - SSH DH Public Key: Z - SSH ECDH Client Public Key: Z - SSH DH Client Public Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - MACsec PSK: Z - MACsec SAK: Z - MACsec KEK: Z - MACsec ICK: Z Root - SSH Private Host Key: Z - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - User Password: Z - CO Password: E,Z - HMAC_DRBG V value: Z - HMAC_DRBG Key value: Z - HMAC_DRBG entropy input: Z - HMAC_DRBG seed: Z - ECDH Shared Secret: Z - DH Shared

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Secret: Z - HMAC Key: Z - SSH Public Host Key: Z - User Authentication Public Keys: Z - CO Authentication Public Keys: Z - JuniperRootCA: Z - PackageCA: Z - SSH ECDH Public Key: Z - SSH DH Public Key: Z - SSH ECDH Client Public Key: Z - SSH DH Client Public Key: Z - MACsec PSK: Z - MACsec SAK: Z - MACsec KEK: Z - MACsec ICK: Z
Perform approved security function	Initiate SSH connection for SSH monitorin	Global Approved Mode indicato	Authentication data (Username and password/p	SSH session	KAS1 KAS2 KTS1 ECDSA SigVer2	Super-user - SSH Private Host Key: E - SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
s (SSH connection)	g and control (CLI)	r "fips" at the CLI combined with successful completion of each service	ublic-key based authentication)		DRBG Entropy Source ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen RSA SigVer Password Hash	ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: G - SSH DH Public Key: G,E,Z - SSH ECDH Public Key: G,E,Z - CO Password: E - CO Authenticati

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						on Public Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Root - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - CO Password: E - CO Authentication Public Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Operator - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG entropy input: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public Keys: E - HMAC_DRBG Key value: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Read-only - SSH Private Host Key: E - SSH ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG Key value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E - SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Keys: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z Unauthorised - SSH Private Host Key: E - SSH ECDH Private Key: G,E,Z - SSH DH Private Key: G,E,Z - SSH Session Key: G,E,Z - HMAC_DRBG V value: E - HMAC_DRBG entropy input: E - HMAC_DRBG seed: E - ECDH Shared Secret: G,E,Z - DH Shared Secret: G,E,Z - HMAC Key: G,E,Z - SSH Public Host Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH ECDH Public Key: G,E,Z - SSH DH Public Key: G,E,Z - User Password: E - User Authentication Public Keys: E - HMAC_DRBG Key value: E - SSH ECDH Client Public Key: W,E,Z - SSH DH Client Public Key: W,E,Z
Console Access	Console monitoring and control (CLI)	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Username, password (set system login user <username> class <crypto-officer/user class> operator authentication plaintext-password)	N/A	Password Hash	Super-user - CO Password: E Operator - CO Password: E Read-only - User Password: E Unauthorised - User Password: E Root - CO Password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Perform self-tests (remote reset)	Software initiated reset, performs self-tests on demand via SSH or via a command	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Control input/reset signal (request vmhost reboot); via a command (request system fips self-test)	N/A	KAS1 KAS2 KTS1 DRBG Entropy Source ECDSA KeyGen ECDSA KeyGen2 ECDSA KeyVer ECDSA SigGen RSA KeyGen RSA SigGen Password Hash CASTs on boot	Super-user - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - HMAC_DRBG Key value: G,Z - HMAC_DRBG V value: G,Z - HMAC_DRBG entropy input: G,Z - HMAC_DRBG seed: G,Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: G,E,Z - SSH ECDH Public Key: G,E - SSH DH Public Key: G,E - CO Password: E - SSH Private Host Key: E - SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E Root - SSH ECDH Private Key: Z - SSH DH Private Key: Z - SSH Session Key: Z - HMAC_DRBG Key value: G,Z - HMAC_DRBG V value: G,Z - HMAC_DRBG entropy input: G,Z - HMAC_DRBG seed: G,Z - ECDH Shared Secret: Z - DH Shared Secret: Z - HMAC Key: G,E,Z - SSH ECDH Public Key: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH DH Public Key: G,E - CO Password: E - SSH Private Host Key: E - SSH Public Host Key: E - User Authentication Public Keys: E - CO Authentication Public Keys: E
Perform self-tests (local reset)	Hardware reset or power cycle	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Control input/reset signal	N/A	CASTs on boot	Super-user Root Operator Read-only Unauthorised Unauthenticated
Load Image	Verification and loading of a validated firmware image into the router/switch	Global Approved Mode indicator "fips" at the CLI combined with successful complet	Image, commands	N/A	ECDSA SigVer Password Hash	Super-user - CO Password: E - JuniperRootCA: E - PackageCA: E Root - CO Password:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		ion of each service				E - JuniperRootCA: E - PackageCA: E
Perform approved security functions (MACsec connection)	Initiate MACsec connection	Global Approved Mode indicator "fips" at the CLI combined with successful completion of each service	Commands (set security macsec connectivity-association connectivity-association-name; set security macsec connectivity-association connectivity-association-name pre-shared key)	MACsec session	MACsec Encryption/Decryption KTS2 MACsec Key Derivation	Root - MACsec PSK: W,E - MACsec SAK: G,R,E - MACsec KEK: G,E - MACsec ICK: G,E Super-user - MACsec PSK: W,E - MACsec SAK: G,R,E - MACsec KEK: G,E - MACsec ICK: G,E

Table 14: Approved Services

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Configure security (security relevant)	Security relevant configuration	RSA with key size less than 2048 ECDSA with ed25519 curve EC Diffie-Hellman with ed25519 curve ARCFOUR Blowfish CAST DSA (SignGen, SigVer, non-compliant) HMAC-MD5 HMAC-RIPEMD160	Root, Super-user

Name	Description	Algorithms	Role
		UMAC TDES	
Perform approved security functions (SSH connection)	Initiate SSH connection for SSH monitoring and control (CLI)	RSA with key size less than 2048 ECDSA with ed25519 curve EC Diffie-Hellman with ed25519 curve ARCFOUR Blowfish CAST DSA (SignGen, SigVer, non-compliant) HMAC-MD5 HMAC-RIPEMD160 UMAC TDES	Root, Super-user, Operator, Read-Only, Unauthorized

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

The module supports loading of firmware from an external source (a complete image replacement) and a firmware load test using ECDSA P-256 with SHA2-256 (CAVP Cert. #A7286) is performed in support of the load. Any firmware/software loaded into this module that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-3 validation.

4.6 Cryptographic Output Actions and Status

The module supports self-initiated cryptographic output in the context of the MACsec protocol and three independent configurations are required serving as three independent internal actions (two actions required at minimum):

- set security macsec connectivity-association <name> cipher-suite
- set interfaces <name> connectivity-association <name>
- set interfaces <name> unit 0 family inet address <ip address>

The following “show” commands indicate the status of the MACsec service:

- show security macsec connections
- show security mka sessions
- show security mka statistics

5 Software/Firmware Security

5.1 Integrity Techniques

The module performs the firmware integrity check using ECDSA P-256 with SHA2-256 (CAVP Cert. #A7286). The ECDSA P-256 public key used for signature verification is a non-SSP and stored persistently across reboots in the module's Non-Volatile RAM (NVRAM) and is exempt from zeroisation.

5.2 Initiate on Demand

The operator can initiate the integrity test on demand by rebooting the module.

5.3 Additional Information

The module firmware image is delivered in the form of a pre-compiled tarball (.tgz).

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

The module contains a limited operational environment since it supports loading of firmware from an external source. The Junos OS 23.4R2 operating system is contained within the module, i.e., the tested configurations listed in the Tested Module Identification – Hardware in this document.

6.2 Configuration Settings and Restrictions

Security rules and restrictions for configuration of the operational environment have been specified in Sections 2.12 and 11.1 of this document.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
The module is completely enclosed in a rectangular nickel or clear zinc coated, cold rolled steel, plated steel and brushed aluminum enclosure; The module enclosure is made of production grade materials; There are no ventilation holes, gaps, slits, cracks, slots,	N/A	N/A

Mechanism	Inspection Frequency	Inspection Guidance
or crevices that would allow for any sort of observation of any component contained within the cryptographic boundary		

Table 16: Mechanisms and Actions Required

The module's physical embodiment is that of a multi-chip standalone device that meets Level 1 Physical Security requirements. No actions are required by the operator to ensure that physical security is maintained.

8 Non-Invasive Security

8.1 Mitigation Techniques

The module does not implement any non-invasive security mitigations and thus the requirements per this section do not apply to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
NVRAM	Non-Volatile Random Access Memory	Static
RAM	Random Access Memory	Dynamic

Table 17: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Entered over SSH - NVRAM	External endpoint	NVRAM	Encrypted	Automated	Electronic	KTS1
Loaded at manufacture	External endpoint	NVRAM	Plaintext	N/A	N/A	
Entered through the CLI via console connection - NVRAM	External endpoint	NVRAM	Plaintext	Manual	Direct	
Output encrypted with MAcsec KEK	RAM	External endpoint (MACsec peer)	Encrypted	Automated	Electronic	KTS2

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Input during SSH negotiation	External endpoint	RAM	Plaintext	Automated	Electronic	
Output during SSH negotiation (host key)	NVRAM	External endpoint	Plaintext	Automated	Electronic	
Output during SSH negotiation (Key Agreement public key)	RAM	External endpoint	Plaintext	Automated	Electronic	

Table 18: SSP Input-Output Methods

The module is complaint with FIPS 140-3 IG 9.5.A MD/DE and AD/EE for SSPs entered via the module's CLI via a direct connection to its serial/console port and for SSPs entered/ouput/established via SSH/MACsec respectively.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroisation command	Command used to zeroise the module: request vmhost zeroize no-forwarding	Used to provide zeroisation as a service	Operator initiated
Power-cycle	Power cycling the module to zeroise temporary SSPs	Power cycling the module to zeroise temporary SSPs	Operator initiated
Session termination	Termination of SSH sessions automatically zeroises temporary SSPs used as part of the session	Termination of SSH sessions automatically zeroises temporary SSPs used as part of the session	Module initiated
Not zeroised	PSP not zeroised since it cannot be modified due to being inaccessible in the filesystem	PSP not zeroised since it cannot be modified due to being inaccessible in the filesystem	N/A
Derivation of SSH session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of SSH session key	EC Diffie-Hellman/Diffie-Hellman shared secrets are zeroised after use in derivation of SSH session key	Module initiated

Table 19: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH Private Host Key	Host key generated, used	P-256 for	Private Host Key - CSP	DRBG Entropy		KAS 1

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	for authentication and encryption in the context of SSH	ECDSA, 2048 bits for RSA - 128 bits for ECDSA, 112 bits for RSA		Source ECDSA KeyGen RSA KeyGen		KAS 2
SSH ECDH Private Key	Ephemeral EC Diffie-Hellman private key used in SSH	KAS-ECC-SSC P-256, P-384, P-512 - 128 bits, 192 bits, 256 bits	ECDH Private Key - CSP	DRBG Entropy Source ECDSA KeyGen2		KAS 1
SSH DH Private Key	Ephemeral Diffie-Hellman private key used in SSH	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	DH Private Key - CSP	DRBG Entropy Source ECDSA KeyGen2		KAS 2
SSH Session Key	SSH Session Key	128 bits, 192 bits, 256 bits - 128 bits, 192 bits, 256 bits	Session Key - CSP		KAS1 KAS2	
User Password	Passwords used to authenticate users to the module	10-20 characters - 1/(96^10) per attempt, 9/(96^10) per minute	User Password - CSP			
CO Password	Passwords used to authenticate COs to the module	10-20 characters - 1/(96^10) per attempt,	CO Password - CSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		9/(96^10) per minute				
HMAC_DRBG V value	A critical value of the internal state of DRBG; per IG D.L Resolution 2	256 bits - 256 bits	Internal state of the DRBG - CSP	DRBG		DRBG
HMAC_DRBG Key value	A critical value of the internal state of DRBG; per IG D.L Resolution 2	440 bits - 440 bits	Internal state of the DRBG - CSP	DRBG		DRBG
HMAC_DRBG entropy input	Entropy input to the HMAC_DRBG; per IG D.L Resolution	512 bits - 448 bits	Entropy input to the HMAC_DRBG - CSP	Entropy Source		
HMAC_DRBG seed	Seed provided to the HMAC_DRBG; per IG D.L Resolution	512 bits - 440 bits	Seed provided to the HMAC_DRBG - CSP	DRBG		DRBG
ECDH Shared Secret	Used in EC Diffie-Hellman (ECDH) exchange	P-256, P-384, P-521 - 128 bits, 192 bits, 256 bits	Shared secret - CSP		KAS1	
DH Shared Secret	Used in Diffie-Hellman (DH) exchange	2048 bits - 112 bits	Shared secret - CSP		KAS2	
HMAC Key	MAC key	128 bits and 256 bits - 128 bits and 256 bits	MAC key - CSP		KAS1 KAS2	
SSH Public Host Key	Host key generated, used to identify the host. Also paired with the private key for authentication and encryption in the context of SSH	P-256 for ECDSA and 2048 bits for RSA - 128 bits for ECDSA, 112 bits for RSA	Public key - PSP	DRBG Entropy Source ECDSA KeyGen RSA KeyGen		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
User Authentication Public Keys	Used to authenticate users to the module	P-256, P-384, P-521 for ECDSA and 2048, 3072 and 4096 bits for RSA - 128, 192, 256 bits for ECDSA, 112, 192 and 256 bits for RSA	Public key - PSP			
CO Authentication Public Keys	Used to authenticate the CO to the module	P-256, P-384, P-521 for ECDSA and 2048, 3072 and 4096 bits for RSA - 128, 192, 256 bits for ECDSA, 112, 192 and 256 bits for RSA	Public key - PSP			
JuniperRoot CA	ECDSA prime256v1 X.509 V3 Certificate Used to verify the validity of the PackagCA	ECDSA P-256 - 128 bits	Public key certificate - Neither			
PackageCA	ECDSA prime256v1 X.509 V3 Certificate	ECDSA P-256 - 128 bits	Public key certificate - Neither			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	Certificate that holds the public key for the signing key used to generate all the signatures used on the packages and signature lists					
SSH ECDH Public Key	Ephemeral EC Diffie-Hellman public key used in SSH	KAS-ECC-SSC P-256, P-384, P-512 - 128 bits, 192 bits, 256 bits for KAS-ECC-SSC	Public key - PSP	DRBG Entropy Source ECDSA KeyGen2		
SSH DH Public Key	Ephemeral Diffie-Hellman public key used in SSH	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	Public key - PSP	DRBG Entropy Source ECDSA KeyGen2		
MACsec PSK	Credential used for device-to-device authentication, consists of the CAK (pre-shared key) and CKN (identifier for the pre-shared key)	128, 256 bits - 128, 256 bits	Symmetric key - CSP			
MACsec SAK	Security Association Key used for creating Security Associations for encryption/decryption of MACsec traffic	128, 256 bits - 128, 256 bits	Symmetric key - CSP	MACsec Key Derivation		
MACsec KEK	Used to transmit SAKs to other	128, 256 bits -	Symmetric key - CSP	MACsec Key		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	members of a MACsec connectivity association	128, 256 bits		Derivation		
MACsec ICK	Used to verify the integrity and authenticity of MACsec protocol data units	128, 256 bits - 128, 256 bits	Symmetric key - CSP	MACsec Key Derivation		
SSH ECDH Client Public Key	Ephemeral EC Diffie-Hellman public key used in SSH (sent by the client to the module acting as the server)	KAS-ECC-SSC P-256, P-384, P-512 - 128 bits, 192 bits, 256 bits for KAS-ECC-SSC	Public key - PSP			
SSH DH Client Public Key	Ephemeral Diffie-Hellman public key used in SSH (sent by the client to the module acting as the server)	2048 bits for KAS-FFC-SSC - 112 bits for KAS-FFC-SSC	Public key - PSP			

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH Private Host Key		NVRAM:Plaintext		Zeroisation command	
SSH ECDH Private Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Private Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH Session Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
User Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated		Zeroisation command	
CO Password	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Obfuscated		Zeroisation command	
HMAC_DRBG V value		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG Key value		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG entropy input		RAM:Plaintext	Until power-cycle	Power-cycle	
HMAC_DRBG seed		RAM:Plaintext	Until power-cycle	Power-cycle	
ECDH Shared Secret		RAM:Plaintext	Until SSH session key derivation	Zeroisation command Power-cycle Derivation of SSH session key	
DH Shared Secret		RAM:Plaintext	Until SSH session key derivation	Zeroisation command Power-cycle Derivation of SSH session key	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
HMAC Key		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH Public Host Key	Output during SSH negotiation (host key)	NVRAM:Plaintext		Zeroisation command	
User Authentication Public Keys	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Zeroisation command	
CO Authentication Public Keys	Entered over SSH - NVRAM Entered through the CLI via console connection - NVRAM	NVRAM:Plaintext		Zeroisation command	
JuniperRootCA	Loaded at manufacture	NVRAM:Plaintext		Not zeroised	
PackageCA	Loaded at manufacture	NVRAM:Plaintext		Not zeroised	
SSH ECDH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Public Key	Output during SSH negotiation (Key Agreement public key)	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
MACsec PSK	Entered over SSH - NVRAM Entered through the	NVRAM:Plaintext		Zeroisation command	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	CLI via console connection - NVRAM				
MACsec SAK	Output encrypted with MAcsec KEK	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
MACsec KEK		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
MACsec ICK		RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH ECDH Client Public Key	Input during SSH negotiation	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	
SSH DH Client Public Key	Input during SSH negotiation	RAM:Plaintext	Until session termination	Zeroisation command Power-cycle Session termination	

Table 21: SSP Table 2

Please Note: The ESV Cert. #E215 corresponds to the module's entropy source: https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E215_PublicUse.pdf. The entropy source does not require any configuration.

9.5 Transitions

Per the NIST SP 800-133Ar2/3 and the programmatic transitions defined by the CMVP, the following algorithm transitions apply to the module, and the algorithms have been designated allowed/non-approved accordingly in Section 2.5:

a. Usage of SHA-1 for SigVer is allowed for legacy use only until 2030. Thereafter, all usage of SHA-1 will be considered a non-approved, not allowed algorithm.
b. Until January 1, 2031, the following algorithms will be considered deprecated:
    a. Hash function and HMAC using SHA-1 hash function

b. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation

c. As of January 1, 2031, the following algorithms will be considered deprecated/disallowed (i.e. non-approved, not allowed)/legacy use:
   a. Use of the 112-bit security strength for classical digital signature and key-establishment mechanisms (deprecated)
   b. Use of the 112-bit security strength for block ciphers (disallowed)
   c. Use of a security strength less than 128-bits but greater than 112 bits for ECDA KeyGen and RSA KeyGen (PKCS #1 v1.5 & PSS) (deprecated)
   d. HMAC using SHA-1 hash function (legacy use)
   e. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Generation (disallowed)
   f. Use of a security strength less than 128-bits but greater than 112 bits for HMAC Verification (legacy use)

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Firmware Integrity Test	Using ECDSA P-256 with SHA2-256	KAT	SW/FW Integrity	"FIPS Self-tests Passed"	Verify

Table 22: Pre-Operational Self-Tests

The module is complaint with FIPS 140-3 IG 10.2.A in that it performs a self-test, a Known Answer Test (KAT) for the ECDSA P-256 (with SHA2-256) algorithm used in the firmware integrity test on each boot prior to executing the firmware integrity test.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG (A6620)	Prediction Resistance: Yes Supports Reseed Capabilities: Mode: SHA2-256 Entropy Input: 256 Nonce: 128 Personalization String Length: 0-	KAT	CAST	NIST 800-90 HMAC DRBG Known Answer Test : Passed	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	256 Increment 8 Additional Input: 8-256 Increment 8 Returned Bits: 1024					
HMAC-SHA2-256 (A6620)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
AES-CBC (A7286)-Encrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A7286) - Encrypt - 192 bits	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A7286) - Encrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Encrypt	During boot
AES-CBC (A7286) - Decrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
AES-CBC (A7286) - Decrypt - 192 bits	Key Length: 192 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
AES-CBC (A7286) - Decrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES-CBC Known Answer Test : Passed	Decrypt	During boot
HMAC-SHA-1 (A7286)	Key Length: 160 bits	KAT	CAST	HMAC-SHA-1 Known Answer	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Test : Passed		
HMAC-SHA2-256 (A7286)	Key Length: 256 bits	KAT	CAST	HMAC-SHA2-256 Known Answer Test : Passed	N/A	During boot
HMAC-SHA2-512 (A7286)	Key Length: 512 bits	KAT	CAST	HMAC-SHA2-512 Known Answer Test : Passed	N/A	During boot
KAS-ECC-SSC Sp800-56Ar3 (A7286) - P256	Domain Parameter Generation Methods: P-256	KAT	CAST	KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed	N/A	During boot
KAS-ECC-SSC Sp800-56Ar3 (A7286) - P384	Domain Parameter Generation Methods: P-384	KAT	CAST	KAS-ECC-EPHEM-UNIFIED-NOKC Known Answer Test: Passed	N/A	During boot
KAS-FFC-SSC Sp800-56Ar3 (A7286)	Domain Parameter Generation Methods: MODP-2048	KAT	CAST	KAS-FFC-EPHEM-NOKC Known Answer Test: Passed	N/A	During boot
KDF SSH (A7286)	Cipher: AES-128, AES-192, AES-256 ; Hash Algorithm: SHA-1, SHA2-256, SHA2-512	KAT	CAST	KDF-SSH-SHA2-256 Known Answer Test: Passed	N/A	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigGen (FIPS186-5) (A7286)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-SIGN Known Answer Test: Passed	Sign	During boot
RSA SigVer (FIPS186-5) (A7286)	Modulus 2048 bits SHA2-256	KAT	CAST	RSA-VERIFY Known Answer Test: Passed	Verify	During boot
ECDSA SigGen (FIPS186-5) (A7286)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	CAST	ECDSA-SIGN Known Answer Test: Passed	Sign	During boot
ECDSA SigVer (FIPS186-5) (A7286)	Curve: P-256 Hash Algorithm: SHA2-256	KAT	CAST	ECDSA-VERIFY Known Answer Test: Passed	Verify	During boot
SHA2-512 (A6550)	SHA2-512	KAT	CAST	SHA-2-512 Known Answer Test: Passed	N/A	During boot
Entropy Test - RCT	NIST SP 800-90B Repetitive Count Test	RCT	CAST	pass	Cutoff value C = 21	During boot and continually
Entropy test - APT	NIST SP 800-90B Adapative Proportion Test	APT	CAST	pass	W = 512; Cutoff value C = 311	During boot and continually
ECDSA KeyGen (FIPS 186-5) - PCT	Curve: P-256 Hash Algorithm: SHA2-256	PCT	PCT	0	Key pair generated for SSP agreement and signature generation/verification in the context of SSHv2 protocol	On key generation
KAS-FFC-SSC	Capabilities: Domain	PCT	PCT	0	Key pair generated for SSP agreement	On key generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Sp800-56Ar3 (A7286) - PCT	Parameter: MODP2048				in the context of SSHv2 protocol	
RSA KeyGen (FIPS186-5) (A7286)	Modulus: 2048 Hash SHA2-256	PCT	PCT	0	Key pair generated for signature generation/verification in the context of SSHv2 protocol	On key generation
AES-KW (A6551) - Encrypt	Key Length: 128 bits	KAT	CAST	AES-KEYWRAP Known Answer Test: Passed	Encrypt	During boot
AES-KW (A6551) - Decrypt	Key Length: 128 bits	KAT	CAST	AES-KEYWRAP Known Answer Test: Passed	Decrypt	During boot
KDF SP800-108 (A6551)	Mode: Counter	KAT	CAST	KBKDF Known Answer Test:Passed	N/A	During boot
AES-GCM (AES 4545) - Encrypt	Key Length: 256 bits	KAT	CAST	0	Encrypt	During boot
AES-GCM (AES 4545) - Decrypt	Key Length: 256 bits	KAT	CAST	0	Decrypt	During boot
AES-CMAC (A6551) - Encrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES128-CMAC Known Answer Test: Passed	Encrypt	During boot
AES-CMAC (A6551) - Decrypt - 128 bits	Key Length: 128 bits	KAT	CAST	AES128-CMAC Known Answer	Decrypt	During boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				Test: Passed		
AES-CMAC (A6551) - Encrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES256-CMAC Known Answer Test: Passed	Encrypt	During boot
AES-CMAC (A6551) - Decrypt - 256 bits	Key Length: 256 bits	KAT	CAST	AES256-CMAC Known Answer Test: Passed	Decrypt	During boot
Firmware Load Test	Curve: P-256 Hash Algorithm: SHA2-256	KAT	SW/FW Load	Host OS upgrade staged. Reboot the system to complete installation !	Verify	On loading of firmware from an external source
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's console (serial) interface	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's console (serial) interface	Manual Entry	Command prompt with "fips" string provided post completion of the test	N/A	On configuration of operator passwords and MACsec PSK

Table 23: Conditional Self-Tests

Cryptographic Algorithm Self-tests (CASTs) are performed on each boot of the module. Other conditional self-tests are performed by the module when the corresponding condition is met. The pairwise consistency tests are performed on key pair generation for use in signature generation/verification (ECDSA and/or RSA tests) and/or for use in KAS-ECC-SSC or KAS-

FFC-SSC SSP agreement (ECDSA and FFC tests respectively). The firmware load test is performed when a firmware image is loaded onto the module from an external source.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Firmware Integrity Test	KAT	SW/FW Integrity	On Demand	Manually via a reboot, via a command (request system fips self-test)

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A6620)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
HMAC-SHA2-256 (A6620)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CBC (A7286)-Encrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CBC (A7286) - Encrypt - 192 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CBC (A7286) - Encrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CBC (A7286) - Decrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CBC (A7286) -	KAT	CAST	On Demand	Manually via a reboot, via a command

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Decrypt - 192 bits				(request system fips self-test)
AES-CBC (A7286) - Decrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
HMAC-SHA-1 (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
HMAC-SHA2-256 (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
HMAC-SHA2-512 (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
KAS-ECC-SSC Sp800-56Ar3 (A7286) - P256	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
KAS-ECC-SSC Sp800-56Ar3 (A7286) - P384	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
KAS-FFC-SSC Sp800-56Ar3 (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
KDF SSH (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
RSA SigGen (FIPS186-5) (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
ECDSA SigGen (FIPS186-5) (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
ECDSA SigVer (FIPS186-5) (A7286)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
SHA2-512 (A6550)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
Entropy Test - RCT	RCT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
Entropy test - APT	APT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
ECDSA KeyGen (FIPS 186-5) - PCT	PCT	PCT	On Demand	Manually via a reboot, via a command (request system fips self-test)
KAS-FFC-SSC Sp800-56Ar3 (A7286) - PCT	PCT	PCT	On Demand	Manually via a reboot, via a command (request system fips self-test)
RSA KeyGen (FIPS186-5) (A7286)	PCT	PCT	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-KW (A6551) - Encrypt	KAT	CAST	On Demand	Manually via a reboot, via a command

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
				(request system fips self-test)
AES-KW (A6551) - Decrypt	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
KDF SP800-108 (A6551)	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-GCM (AES 4545) - Encrypt	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-GCM (AES 4545) - Decrypt	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CMAC (A6551) - Encrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CMAC (A6551) - Decrypt - 128 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CMAC (A6551) - Encrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
AES-CMAC (A6551) - Decrypt - 256 bits	KAT	CAST	On Demand	Manually via a reboot, via a command (request system fips self-test)
Firmware Load Test	KAT	SW/FW Load	On Demand	Manually via loading of firmware from an external source

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Manual entry test (duplicate entries)	Duplicate entry test required for entry of operator passwords and MACsec PSK via direct connection to the module's console (serial) interface	Manual Entry	On Demand	Manually via configuration of operator passwords and MACsec PSK

Table 25: Conditional Periodic Information

The pre-operational firmware integrity test as well as all CASTs must be completed successfully prior to any other use of cryptography by the module in the Approved mode of operation. These tests can also be performed periodically by rebooting the module or via a command (request system fips self-test).

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Hard Error state	If the pre-operation firmware integrity test, if any of the CASTs or pair-wise consistency tests fail, then the module returns an error indicator, inhibits all data output and enters the hard error state	If the pre-operational firmware integrity test or if any of the CASTs fail	N/A	"FIPS error: self-test failure" for firmware integrity failure, "FIPS error 1: <name of the algorithm> Known Answer Test: Failed" for CAST failure and -1 for pair-wise consistency test failure
Soft Error state	*In case of a firmware load test failure, the module rejects the firmware, returns an error indicator and enters the soft error state *In the event of an APT or RCT health test failure, output from the entropy source is inhibited, all entropy accumulated in the conditioning context is discarded and the start-up	If the firmware load test fails If the APT or RCT test fails	N/A for firmware load test failure; In case of APT and/or RCT failures, new data continues to be tested by the health tests, and once both health tests indicate a "pass", the entropy source again outputs data	veriexec: cannot verify <img_file>.esig: ERROR: Failed loading signature file <path/to/signature>; entropy data discarded in case of APT/RCT failure

Name	Description	Conditions	Recovery Method	Indicator
	health-tests are performed again			

Table 26: Error States

If the pre-operation firmware integrity test or if any of the CASTs fail, then the module returns the error indicator “FIPS error: self-test failure”, inhibits all data output and enters the hard error state.

If the conditional self-tests fail, the module enters the soft error state, i.e., it rejects the generated keypair/loaded image, returns an error indicator and resumes normal operation.

10.5 Operator Initiation of Self-Tests

Each time the module is powered up it tests that all the cryptographic algorithms operate correctly, and that sensitive data have not been damaged. Pre-operational as well as Conditional Cryptographic Algorithm Self-tests (CAST) are performed on each power up/boot of the module and on demand by power cycling the module (Perform self-tests (remote reset) service) or via a command (request system fips self-test).

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Crypto Officer must follow the procedures defined below for secure installation, initialization, startup and operation of the module.

Crypto Officer Guidance
The Crypto Officer must check to verify the firmware image being loaded on the module is the FIPS 140-3 validated version/image. If the image is the FIPS 140-3 validated image, then proceed with installation of the image.

Installing The Firmware Image
Download the validated firmware image from https://www.juniper.net/support/downloads/junos.html. Log in to the HPE Juniper Networking authentication system using the username (generally your e-mail address) and password supplied by HPE Juniper Networking representatives. Select the validated firmware image. Download the firmware image to a local host or to an internal software distribution site.

Connect to the console port on the device from your management device and log in to the Junos OS CLI. Copy the firmware package to the device to the /var/tmp/ directory. Install the new package on the device using the following command: operator > request system software add /var/tmp/<package>.tgz.

NOTE: If you need to terminate the installation, do not reboot your device; instead, finish the installation and then issue the request system software delete package.tgz command, where package.tgz is, for example, jinstall-host-qfx-10-f-x86-64.23.4R2.secure-signed.tgz.This is your last chance to stop the installation.

Reboot the device to complete the load and start the installation:
operator> request system reboot

After the reboot has completed, log in and use the show version command to verify that the new version of the firmware is successfully installed.

Also install the built-in fips-mode.tgz package needed for enabling the Approved-mode and the jpfe-fips package needed for execution of the CASTs. Please note that this is a one-time installation post which the module remains in the Approved mode once enabled and automatically executes the CASTs on each boot without requiring any operator or external intervention. The following are the commands used for installing these packages:

operator>request system software add optional://fips-mode.tgz
operator>request system software add optional://jpfe-fips.tgz

Enabling Approved Mode of Operation
The Crypto Officer is responsible for initializing the module in the Approved mode of operation. The Approved mode of operation is not automatically enabled. The Crypto Officer shall place the module in the Approved mode by first zeroising it to ensure no SSPs are present. Next, the cryptographic officer shall follow the steps found in the Junos OS FIPS Evaluated Configuration Guide for QFX Series, Release 23.4R2 document Chapter 2 to place the module into an Approved mode of operation. The steps from the aforementioned document have been reiterated below.

To enable the Approved mode in Junos OS on the module:

1. Zeroise the module using the “request system zeroize” command. Zeroise the module using the “request system zeroize” command. The Crypto Officer shall retain control of the module while zeroisation is in process. Once the module comes up in the “amnesiac mode” post zeroisation, connect to it using the console port with username “root” and enter the configuration mode. Enable the Approved mode on the module by setting the Approved level to 1, and verify the level:

[edit]
root# set system fips chassis level 1

[edit]

root# show system fips chassis level
level 1;

2. Configure the root-authentication password (i.e., Crypto Officer credentials) as follows:
root> edit
Entering configuration mode

[edit]
root# set system root-authentication plain-text-password
New password:
Retype new password:

3. Commit the configuration
[edit]
root# commit
configuration check succeeds
   Generating RSA key /etc/ssh/fips_ssh_host_key
   Generating RSA2 key /etc/ssh/fips_ssh_host_rsa_key
   Generating ECDSA key /etc/ssh/fips_ssh_host_ecdsa_key
 'system' reboot is required to transition to fips level 1
   commit complete

4. Reboot the device:
 [edit]
root# run request system reboot
Reboot the system ? [yes,no] (no) yes
 During the reboot, the device runs the pre-operational firmware integrity test and all CASTs. It returns a login prompt.

5. After the reboot has completed, log in and use the show version command to verify the firmware version is the validated version:
root:fips > show version

Placing the Module in the Non-Approved Mode of Operation
As Crypto Officer, the operator needs to disable the Approved mode of operation on the device to return it to the non-Approved mode of operation. To disable the Approved mode on the device, the module must be zeroised (step 1 defined above).

11.2 Administrator Guidance

For further information and for the Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for HPE Juniper Networking QFX, Release 23.4R2 document.

11.3 Non-Administrator Guidance

For further information and for the non-Administrator guidance, please see the Junos OS FIPS Evaluated Configuration Guide for HPE Juniper Networking QFX, Release 23.4R2 document.

11.4 Maintenance Requirements

No other maintenance requirements apply for operation of the module in the Approved/non-Approved modes as defined above.

11.5 End of Life

The module can be securely sanitized at the end of its lifetime by zeroising it.

12 Mitigation of Other Attacks

12.1 Attack List

The module does not implement any mitigation of other attacks and thus the requirements per this section do not apply to the module.