HPE Juniper Networking

FIPS 140-3 Non-Proprietary Security Policy

HPE Juniper Networking EX4400 Ethernet Switch

Version: 24.2R2-S3.3

Prepared by:



www.lightshipsec.com

Prepared for:



Hewlett Packard Enterprise

HPE Juniper Networking
1133 Innovation Way
Sunnyvale, California 94089
USA
408.745.2000
1.888 JUNIPER
https://www.hpe.com/us/en/networking/hpe-juniper-networking.html

Table of Contents

1 General 6
1.1 Overview 6
1.2 Security Levels 6
2 Cryptographic Module Specification 6
2.1 Description 6
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.3 Excluded Components 9
2.4 Modes of Operation 9
2.5 Algorithms 10
2.6 Security Function Implementations 14
2.7 Algorithm Specific Information 16
2.8 RBG and Entropy 17
2.9 Key Generation 17
2.10 Key Establishment 17
2.11 Industry Protocols 18
3 Cryptographic Module Interfaces 18
3.1 Ports and Interfaces 18
4 Roles, Services, and Authentication 19
4.1 Authentication Methods 19
4.2 Roles 19
4.3 Approved Services 19
4.4 Non-Approved Services 24
4.5 External Software/Firmware Loaded 24
5 Software/Firmware Security 24
5.1 Integrity Techniques 24
5.2 Initiate on Demand 24
6 Operational Environment 25
6.1 Operational Environment Type and Requirements 25
6.2 Configuration Settings and Restrictions 25
7 Physical Security 25
7.1 Mechanisms and Actions Required 25
8 Non-Invasive Security 25
9 Sensitive Security Parameters Management 25
9.1 Storage Areas 25

9.2 SSP Input-Output Methods 26
9.3 SSP Zeroization Methods 26
9.4 SSPs 27
9.5 Transitions 31
10 Self-Tests 32
10.1 Pre-Operational Self-Tests 32
10.2 Conditional Self-Tests 32
10.3 Periodic Self-Test Information 34
10.4 Error States 36
10.5 Operator Initiation of Self-Tests 37
11 Life-Cycle Assurance 37
11.1 Installation, Initialization, and Startup Procedures 37
11.2 Administrator Guidance 37
11.2.1Installing the Junos OS firmware image 37
11.2.2 Configure the device for the Approved mode 38
11.2.3 Administrator Module Operational Rules 38
11.2.4 Zeroizing the System 39
11.3 Non-Administrator Guidance 39
11.4 Design and Rules 39
11.4.1 Module Design Rules 39
11.5 End of Life 39
12 Mitigation of Other Attacks 40

List of Tables

Table 1: Security Levels 6
Table 2: Tested Module Identification – Hardware 9
Table 3: Modes List and Description 10
Table 4: Approved Algorithms 13
Table 5: Vendor-Affirmed Algorithms 13
Table 6: Security Function Implementations 16
Table 7: Entropy Certificates 17
Table 8: Entropy Sources 17
Table 9: Ports and Interfaces 18
Table 10: Authentication Methods 19
Table 11: Roles 19
Table 12: Approved Services 24
Table 13: Storage Areas 26
Table 14: SSP Input-Output Methods 26
Table 15: SSP Zeroization Methods 26
Table 16: SSP Table 1 29
Table 17: SSP Table 2 31
Table 18: Pre-Operational Self-Tests 32
Table 19: Conditional Self-Tests 34
Table 20: Pre-Operational Periodic Information 34
Table 21: Conditional Periodic Information 36
Table 22: Error States 37

List of Figures

Figure 1 – EX4400-24P Ethernet switch (front) 7
Figure 2 –EX4400-24P Ethernet switch (rear) 7
Figure 3 – EX4400-24MP Ethernet switch (front) 7
Figure 4 – EX4400-24MP Ethernet switch (rear) 7
Figure 5 –EX4400-24T Ethernet switch (front) 7
Figure 6 – EX4400-24T Ethernet switch (rear) 7
Figure 7 – EX4400-24X Ethernet switch (front) 7
Figure 8 – EX4400-24X Ethernet switch (rear) 7
Figure 9 – EX4400-48T Ethernet switch (front) 8
Figure 10 – EX4400-48T Ethernet switch (rear) 8
Figure 11 – EX4400-48P Ethernet switch (front) 8
Figure 12 – EX4400-48P Ethernet switch (rear) 8
Figure 13 – EX4400-48XP Ethernet switch (front) 8
Figure 14 – EX4400-48XP Ethernet switch (rear) 8
Figure 15 – EX4400-48F Ethernet switch (front) 8
Figure 16 – EX4400-48F Ethernet switch (rear) 8
Figure 17 – EX4400-48MP Ethernet switch (front) 8
Figure 18 – EX4400-48MP Ethernet switch (rear) 9
Figure 19 – EX4400-48MXP Ethernet switch (front) 9
Figure 20 – EX4400-48MXP Ethernet switch (rear) 9

This document may be reproduced and distributed only in its original entirety without revision.

1 General

1.1 Overview

This is a non-proprietary Cryptographic Module Security Policy for the HPE Juniper Networking EX4400 Ethernet Switch, running firmware version 24.2R2-S3.3.

1.2 Security Levels

The HPE Juniper Networking EX4400 Ethernet Switch is validated at the FIPS 140-3 section levels shown in Table 1.

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	2
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

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The cryptographic module runs Junos OS, HPE Juniper Networking's reliable, high-performance, and modular network operating system, which is supported across all of HPE Juniper Networking's physical and virtual switching and security platforms.

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 devices.

Module Type:
Hardware

Module Embodiment:
Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary is defined as the outer edge of the chassis. The chassis is a rigid sheet-metal structure that houses all components of the device.

The physical form of the module is depicted below.



Figure 1 – EX4400-24P Ethernet switch (front)



Figure 2 –EX4400-24P Ethernet switch (rear)



Figure 3 – EX4400-24MP Ethernet switch (front)



Figure 4 – EX4400-24MP Ethernet switch (rear)



Figure 5 –EX4400-24T Ethernet switch (front)



Figure 6 – EX4400-24T Ethernet switch (rear)



Figure 7 – EX4400-24X Ethernet switch (front)



Figure 8 – EX4400-24X Ethernet switch (rear)



Figure 9 – EX4400-48T Ethernet switch (front)



Figure 10 – EX4400-48T Ethernet switch (rear)



Figure 11 – EX4400-48P Ethernet switch (front)



Figure 12 – EX4400-48P Ethernet switch (rear)



Figure 13 – EX4400-48XP Ethernet switch (front)



Figure 14 – EX4400-48XP Ethernet switch (rear)



Figure 15 – EX4400-48F Ethernet switch (front)



Figure 16 – EX4400-48F Ethernet switch (rear)



Figure 17 – EX4400-48MP Ethernet switch (front)



Figure 18 – EX4400-48MP Ethernet switch (rear)

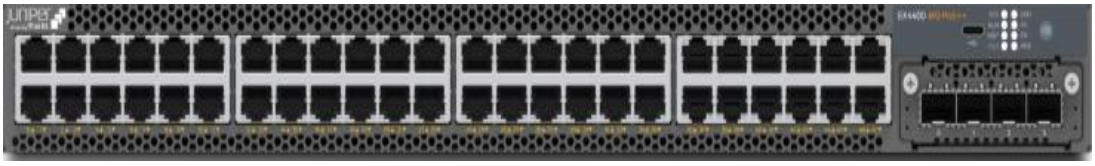


Figure 19 – EX4400-48MXP Ethernet switch (front)



Figure 20 – EX4400-48MXP Ethernet switch (rear)

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
EX4400-24P	EX4400-24P	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-24MP	EX4400-24MP	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-24T	EX4400-24T	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-24X	EX4400-24X	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48T	EX4400-48T	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48P	EX4400-48P	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48XP	EX4400-48XP	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48F	EX4400-48F	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48MP	EX4400-48MP	24.2R2-S3.3	Intel Atom C3558R (Denverton)	
EX4400-48MXP	EX4400-48MXP	24.2R2-S3.3	Intel Atom C3558R (Denverton)	

Table 2: Tested Module Identification – Hardware

2.3 Excluded Components

No components are excluded from the requirements of FIPS PUB 140-3.

2.4 Modes of Operation

When installed, initialized and configured as specified in Section 11 of the Security Policy, the module only operates in the approved mode of operation.

Mode Name	Description	Type	Status Indicator
Approved	Approved Mode	Approved	<user>@<hostname>:fips>

Table 3: Modes List and Description

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6928	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A6931	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A6931	Key Length - 128, 192, 256 Tag Length - 112, 128, 32, 48, 64, 80, 96 IV Length - IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length - Payload Length: 0-256 Increment 8 AAD Length - AAD Length: 0, 256, 65536	SP 800-38C
AES-CMAC	A6928	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	A6931	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-CTR	AES 4544	Key Length - 128, 256 Counter Source - External	SP 800-38A
AES-CTR	AES 4545	Key Length - 128, 256 Counter Source - External	SP 800-38A
AES-CTR	AES 4550	Key Length - 128, 256 Counter Source - External	SP 800-38A
AES-CTR	C996	Direction - Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	A6928	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A6931	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	AES 4544	Direction - Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	AES 4545	Direction - Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	AES 4550	Direction - Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	C996	Direction - Encrypt Key Length - 128, 256	SP 800-38A
AES-GCM	AES 4544	Direction - Decrypt, Encrypt Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 64, 96 IV Length - IV Length: 96	SP 800-38D

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 120, 128 AAD Length - AAD Length: 120, 128	
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-GCM	AES 4550	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-GCM	C996	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 128, 120 AAD Length - AAD Length: 128, 120	SP 800-38D
AES-KW	A6928	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128 Payload Length - Payload Length: 128, 192, 256, 320, 4096	SP 800-38F
AES-XPN	AES 4544	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
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
AES-XPN	AES 4550	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
AES-XPN	C996	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128, 120 AAD Length - AAD Length: 128, 120 Tag Length - 104, 112, 120, 128, 64, 96 IV Generation - External Salt Generation - External	SP 800-38D
DSA KeyGen (FIPS186-4)	A6931	L - 2048, 3072 N - 224, 256	FIPS 186-4
ECDSA KeyGen (FIPS186-5)	A6931	Curve - P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyVer (FIPS186-5)	A6931	Curve - P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A6931	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A6931	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC DRBG	A6929	Prediction Resistance - No 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-SHA2-224	A6931	MAC - MAC: 224 Key Length - Key Length: 224	FIPS 198-1
HMAC-SHA2-256	A6929	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A6930	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-256	A6931	MAC - MAC: 256 Key Length - Key Length: 256	FIPS 198-1
HMAC-SHA2-512	A6931	MAC - MAC: 512 Key Length - Key Length: 512	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6931	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
KAS-FFC-SSC Sp800-56Ar3	A6931	Domain Parameter Generation Methods - FC, MODP-2048 Hash Function Z - SHA2-256 Scheme - dhEphem - KAS Role - responder	SP 800-56A Rev. 3
KDF SP800-108	A6928	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)	A6931	Cipher - AES-128, AES-192, AES-256, TDES Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A6931	Key Generation Mode - probable Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Fixed Public Exponent - 010001 Info Generated By Server - No	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
		Private Key Format - standard Public Exponent Mode - fixed	
RSA SigGen (FIPS186-5)	A6931	Hash Pair - Hash Algorithm - SHA2-256 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-5)	A6931	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	A6931	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
Safe Primes Key Verification	A6931	Safe Prime Groups - MODP-2048	SP 800-56A Rev. 3
SHA2-224	A6931	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A6929	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A6930	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A6931	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A6929	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A6931	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6929	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6930	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A6931	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG - OpenSSL	Key Type:Asymmetric	Junos 24.2R2 - openSSL	SP 800-133 Rev.2 Section 4 example 1 - Using the Output of a Random Bit Generator

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:
N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

Legacy Usage - The module supports the following implementations for legacy use/support per NIST SP 800-131Ar2:

2.6 Security Function Implementations

The module implements the security functions listed in the following table.

Name	Type	Description	Properties	Algorithms
KAS-SSC (SSH)	KAS-SSC	Key Agreement Scheme Shared Secret Computation for SSH		KAS-ECC-SSC Sp800-56Ar3: (A6931) KAS-FFC-SSC Sp800-56Ar3: (A6931) Safe Primes Key Generation: (A6931) Safe Primes Key Verification: (A6931) DSA KeyGen (FIPS186-4): (A6931)
KeyGen (SSH)	AsymKeyPair-KeyGen AsymKeyPair-KeyVer	Key Generation used for SSH authentication keys		ECDSA KeyGen (FIPS186-5): (A6931) RSA KeyGen (FIPS186-5): (A6931) HMAC DRBG: (A6929) DSA KeyGen (FIPS186-4): (A6931) Safe Primes Key Generation: (A6931) Safe Primes Key Verification: (A6931)
SigGen (SSH)	DigSig-SigGen	Signature Generation for peer authentication in SSH		ECDSA SigGen (FIPS186-5): (A6931) RSA SigGen (FIPS186-5): (A6931) SHA2-224: (A6931) SHA2-384: (A6931) SHA2-512: (A6931)
SigVer (SSH)	DigSig-SigVer	Signature Verification for peer authentication in SSH		ECDSA SigVer (FIPS186-5): (A6931) SHA2-256: (A6931) SHA2-384: (A6931)

Name	Type	Description	Properties	Algorithms
				SHA2-512: (A6931) RSA SigVer (FIPS186-5): (A6931)
MAC (SSH)	MAC	Message authentication for SSH		HMAC-SHA2-224: (A6931) HMAC-SHA2-512: (A6931) SHA2-256: (A6931) SHA2-512: (A6931) SHA2-224: (A6931)
KAS KeyGen (SSH)	KAS-KeyGen	Key Generation for Key Agreement in SSH		ECDSA KeyGen (FIPS186-5): (A6931) HMAC DRBG: (A6929)
KDF (SSH)	KAS-135KDF	Key derivation function for SSH		KDF SSH: (A6931) SHA2-256: (A6931) SHA2-384: (A6931) SHA2-512: (A6931)
Full KAS (SSH)	KAS-Full	Full Key Agreement for SSH	KAS-ECC:SP 800-56A Rev. 3 (FIPS 140-3 IG D.F Scenario 2, path 2). KAS-FFC:SP 800-56A Rev. 3 Section 5.6.2.1.4 (b)	ECDSA KeyGen (FIPS186-5): (A6931) ECDSA KeyVer (FIPS186-5): (A6931) KAS-ECC-SSC Sp800-56Ar3: (A6931) SHA2-256: (A6931) SHA2-384: (A6931) SHA2-512: (A6931) KDF SSH: (A6931) KAS-FFC-SSC Sp800-56Ar3: (A6931) Safe Primes Key Generation: (A6931) Safe Primes Key Verification: (A6931) DSA KeyGen (FIPS186-4): (A6931)
KTS (SSH)	KTS-Wrap	Key Transport using SSH as per IG D.G provisions	Key Transport: Key Transport using SSH as per IG D.G	AES-CBC: (A6931) AES-CTR: (A6931) HMAC-SHA2-256: (A6931) HMAC-SHA2-512: (A6931)
SHA (libMD)	SHA	Message Digest Generation		SHA2-256: (A6930) SHA2-512: (A6930)
MAC (libMD)	MAC	Message Authentication		HMAC-SHA2-256: (A6930) SHA2-256: (A6930)
DRBG (Kernel)	DRBG	Random Bit Generation		HMAC DRBG: (A6929)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-256: (A6929) SHA2-256: (A6929)
SHA (Kernel)	SHA	Message Digest Generation		SHA2-512: (A6929) SHA2-256: (A6929) SHA2-384: (A6929)
Verify Image	DigSig-SigVer	Verification of firmware image		SHA2-256: (A6931) ECDSA SigVer (FIPS186-5): (A6931)
Key Derivation (MACsec)	KBKDF	Derivation of MACsec MKA keys		KDF SP800-108: (A6928)
Key wrap (MACsec)	KTS-Wrap	Distribution of MACsec SAK	:IG D.G : Approved Bullet 2 / Bit Strength: Provides 128 or 256 bits of encryption strength	AES-KW: (A6928)
Encrypt / Decrypt (MACsec)	BC-UnAuth	Encryption and decryption of MACsec data		AES-CBC: (A6928) AES-ECB: (A6928, AES 4545, AES 4544, AES 4550, C996) AES-XPN: (AES 4545, AES 4544, AES 4550, C996) AES-GCM: (AES 4545, AES 4544, AES 4550, C996) AES-CTR: (AES 4545, AES 4544, AES 4550, C996)
Integrity (MACsec)	MAC	MACsec protocol data integrity protection		AES-CMAC: (A6928)
Symmetric Encryption / Decryption	BC-UnAuth BC-UnAuthDecrypt	Unauthenticated encryption / decryption		AES-CBC: (A6931) AES-CTR: (A6931) AES-CCM: (A6931) AES-ECB: (A6931)
SHA (OpenSSL)	SHA	Message Digest		SHA2-224: (A6931) SHA2-256: (A6931) SHA2-512: (A6931) SHA2-384: (A6931)

Table 6: Security Function Implementations

2.7 Algorithm Specific Information

In reference to the MACsec protocol, the module can take on the role of Peer or Authenticator.

The AES GCM IV construction is performed in compliance with IG C.H scenario 1c (MACsec per IEEE 802.1AE and its amendments). The module shall only be used with CMVP-validated modules when supporting the MACsec protocol for providing Peer, Authenticator functionality.

Conformance to IG C.H 3a - In case the module's power is lost and then restored, a new key for use with the AES-GCM encryption/decryption shall be re-established by the module.

IG C.K Conformance:
Per IG C.K resolution 3, the DSA algorithm implemented by the module are used in KAS-FFC-SSC and is permits FIPS 186-type FFC parameter-size sets (SP 800-56Arev3 Sec. 5.6.1.1.2).

DSA KeyGen (FIPS186-4) is only implemented for use as a part of an approved SP 800-56Ar3 FFC scheme. In accordance with this, only the FIPS 186-type parameter set FC (2048, 256) from SP 800-56Arev3 are supported by the module.

Refer to Security Policy Section 9.5 - Transitions for additional context.

2.8 RBG and Entropy

Cert Number	Vendor Name
E121	Juniper

Table 7: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Junos OS Physical Entropy Source	Physical	Intel Corporation Intel(R) Atom(R) Denverton-16 FCBGA1310 Intel(R) Atom(R) C3958 Processor	128	Full Entropy	A2153

Table 8: Entropy Sources

The entropy source is used to seed the module's HMAC DRBG with the minimum required 384-bits of entropy. Each 128-bit block of conditioned output from the entropy source contains full entropy. The HMAC DRBG is used for all random data required by the module, including key generation.

There are no initialization procedures required by the users of the module to operate the entropy source in a compliant manner. The module complies with the ESV Public Use document of the validated entropy source (Cert. E121).

2.9 Key Generation

The cryptographic module implements the key generation methods listed above in the Security Functions Implementations table.

2.10 Key Establishment

The cryptographic module implements the key establishment methods listed above in the Security Functions Implementations table.

The module complies with FIPS 140-3 IG D.F required key agreement assurances in accordance with Section 5.6.2 of SP800-56A Rev. 3. All the key agreement protocols implemented by the module are Diffie-Hellman based.

2.11 Industry Protocols

The cryptographic module supports the protocols listed below. No part of these protocols, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP. The SSH algorithms allow independent selection of key exchange, authentication, cipher, and integrity. In reference to the supported protocols table below, each column of options for a given protocol is independent and may be used in any viable combination.

Protocol	Key Exchange	Auth	Cipher	Integrity
SSHv2	EC Diffie-Hellman P-256 EC Diffie-Hellman P-384 EC Diffie-Hellman P-521 Diffie-Hellman MODP-2048 Diffie-Hellman FC	ECDSA P-256 ECDSA P-384 ECDSA P-521 MODP-2048 FC RSA 2048 RSA 3072 RSA 4096	AES CBC 128/192/256 AES CTR 128/192/256	HMAC-SHA-256 HMAC-SHA-512
MACsec	MACsec Key Agreement (SP800-108 KDF, AES-CMAC-128/256, AES-KW 128/256)	Shared secret	AES-GCM-128 AES-GCM-256	

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The following table maps each physical interface to one or more logical interface types defined in the FIPS 140-3 standard. The module does not have a Control Output Interface.

Physical Port	Logical Interface(s)	Data That Passes
RJ-45 Management Interfaces, RJ-45 Ethernet Interfaces, Serial Port, USB Port	Data Input	Data input into the module for the services defined in Table - Approved Services
RJ-45 Management Interfaces, RJ-45 Ethernet Interfaces, Serial Port	Data Output	Data output from the module for the services defined in Table - Approved Services
RJ-45 Management Interfaces, RJ-45 Ethernet Interfaces, Serial Port, USB Port	Control Input	Control input into the module for the services defined in Table - Approved Services
RJ-45 Management Interfaces, RJ-45 Ethernet Interfaces, Serial Port	Status Output	Status output from the module for the services defined in Table - Approved Services
Power	Power	Power supply
Alarm LEDs	Status Output	Status indicator lighting
Reset Button	Control Input	Reset Signal

Table 9: Ports and Interfaces

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module implements three forms of role-based authentication methods, as described in the following table.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
SSH - Password	Password length range 10-20 characters	Username and Password over SSH	1/(96^10)	< 1/1,000,000
SSH - ECDSA	SSH using ECDSA login	ECDSA SigVer (FIPS186-5) (A6931)	1/(2^128)	15,000 attempts per minute
SSH - RSA	SSH using RSA login	RSA SigVer (FIPS186-5) (A6931)	2048 bits	< 1/1,000,000

Table 10: Authentication Methods

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	SSH - Password SSH - ECDSA SSH - RSA
User	Role	User	SSH - Password SSH - ECDSA SSH - RSA

Table 11: Roles

The module supports two roles: Crypto Officer (CO) and User. The module supports concurrent operators. The default number of concurrent operators is set to a limit of 10. The module enforces the separation of roles using any of the role-based operator authentication methods in Section 4.1.

The Crypto Officer role configures and monitors the module via a console or SSH connection. The Crypto Officer has permission to view and edit secrets within the module.

The User role monitors the module via the console or SSH. The user role cannot change the configuration.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure	Non-security relevant configuration	none	CLI Command	Status	None	Crypto Officer
Configure Security	Security relevant configuration	Global Indicator, "presence of ":fips" in the command line in combination with successful completion of service	CLI Command	Status	KAS-SSC (SSH) KeyGen (SSH) SigGen (SSH) SigVer (SSH) MAC (SSH) KAS KeyGen (SSH) KDF (SSH) Full KAS (SSH) KTS (SSH) SHA (libMD) MAC (libMD) DRBG (Kernel) SHA (Kernel) Verify Image Key Derivation (MACsec) Key wrap (MACsec) Encrypt / Decrypt (MACsec) Integrity (MACsec) Symmetric Encryption / Decryption SHA (OpenSSL)	Crypto Officer - HMAC_DRBG V value : G,R,W - HMAC DRBG key value: G,R,W - HMAC_DRBG Entropy Input : G,R,W - HMAC_DRBG Seed : G,R,W - SSH DH Shared Secret: W - SSH Private Host Key: W,E - SSH Public Host Key: R,W,E - SSH DH Private Key: R,W,E - SSH DH Public Key: R,W,E - SSH-SEK: W - Login Password: R,W - Root CA: R - Package CA: R - MACsec CAK: G,R,W - MACsec PSK: G,R,W - MACsec CKN: G,R,W - MACsec SAK: G,R,W - MACsec KEK: G,R,W - MACsec ICK: G,R,W - Operator RSA Public Key: R,W,E - ECDH Public Key: R,W,E - ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: R,W,E - ECDH Shared Secret: G,W,E - SSH Integrity Keys: R,W,E
Secure Traffic	MACsec encrypted transfer of data distribution of keys	Successful completion of service	MACsec traffic frames	MACsec traffic frames	Key Derivation (MACsec) Key wrap (MACsec) Encrypt / Decrypt (MACsec) Integrity (MACsec)	Crypto Officer - MACsec CAK: G,E - MACsec PSK: G,E - MACsec KEK: G,E - MACsec SAK: G,R,E - MACsec ICK: G,R
Show Status	Show Status	Global Indicator, "presence of ":fips" in the command line in combination with successful completion of service	CLI command	Status	None	Crypto Officer User
Show Status	Show status	CLI prompt				
Zeroize	Zeroize all CSPs	Successful completion of service	CLI command	Non (completion indicator is implicitly provided by the module rebooting	None	Crypto Officer - HMAC_DRBG V value : Z - HMAC DRBG key value: Z - HMAC_DRBG Entropy Input : Z - HMAC_DRBG Seed : Z - SSH DH Shared Secret: Z - SSH Private Host Key: Z - SSH Public Host Key: Z - SSH DH Private Key: Z - SSH DH Public Key: Z - SSH-SEK: Z - MACsec CAK: Z - MACsec PSK: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- MACsec CKN: Z - MACsec SAK: Z - MACsec KEK: Z - MACsec ICK: Z - Operator RSA Public Key: Z - ECDH Shared Secret: Z - ECDH Public Key: Z - ECDH Private Key: Z - SSH Integrity Keys: Z - Login Password: Z
SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	Successful completion of service	SSH packets	SSH packet status	SigGen (SSH) SigVer (SSH) KAS KeyGen (SSH) KDF (SSH) Full KAS (SSH) KTS (SSH) SHA (libMD) MAC (libMD) DRBG (Kernel) Symmetric Encryption / Decryption	Crypto Officer - HMAC_DRBG V value : G,R - HMAC DRBG key value: G,R - HMAC_DRBG Seed : R - HMAC_DRBG Entropy Input : R - SSH DH Private Key: G,E - SSH DH Public Key: G,E - SSH DH Shared Secret: G,E - SSH Private Host Key: E - SSH Public Host Key: E - Login Password: R - Operator RSA Public Key: R - ECDH Shared Secret: E - ECDH Public Key: G - ECDH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: G - SSH Integrity Keys: G User - HMAC_DRBG V value : G,R - HMAC DRBG key value: G,R - HMAC_DRBG Seed : R - HMAC_DRBG Entropy Input : R - SSH DH Private Key: G,E - SSH DH Public Key: G,E - SSH DH Shared Secret: G,E - SSH Private Host Key: E - Login Password: R - Operator RSA Public Key: R - ECDH Shared Secret: E - ECDH Private Key: G - SSH Integrity Keys: G - ECDH Public Key: G
MACsec connect	Initiate MACsec connection	Successful completion of service	MACsec link configuration CKN,CAK	MACsec frames, Status	Key Derivation (MACsec)	Crypto Officer - MACsec ICK: E - MACsec SAK: E - MACsec KEK: E
Console access	Console monitoring and control	None	CLI command	Status	None	Crypto Officer - Login Password: E User - Login Password: E
Traffic	Traffic requiring no cryptographic services	None	Traffic in	Traffic out	None	

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Load Image	Loading of firmware image	Successful completion of service	CLI command	Status	Verify Image	Crypto Officer - Package CA: R - Root CA: R
Perform self-tests	On demand self-tests	Successful completion of service	CLI command	Status	None	Crypto Officer
Show Version	Show system information identifying module	Version information in combination with the successful completion of service	CLI command	Status	None	Crypto Officer User

Table 12: Approved Services

4.4 Non-Approved Services

The module does not offer any non-approved services.

N/A for this module.

4.5 External Software/Firmware Loaded

The module includes a firmware load service that is used to install the Junos OS firmware image as part of installation of the module, as described in Section 11.1. The loaded firmware is a complete image replacement and constitutes an entirely new module and version of Junos OS which would require a separate FIPS 140-3 validation. The design of the modules ensures that all data output, via the data output interface, is inhibited during firmware loading.

5 Software/Firmware Security

5.1 Integrity Techniques

The cryptographic module implements a firmware integrity self-test that uses ECDSA SigVer P-256 with SHA-256 to ensure the integrity of all Junos OS firmware components. The self-test is automatically run on power-up.

5.2 Initiate on Demand

The firmware integrity test can be run on demand by the module's operator by power cycling the module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

The Module has control over its own SSPs. The operational environment provides the capability to separate individual processes from each other to prevent uncontrolled access to CSPs and uncontrolled modifications of SSPs. This separation is enforced regardless if this data is in the process memory or stored on persistent storage within the operational environment. The module ensures that direct access to SSPs is restricted to the cryptographic module and the trusted parts of the operational environment. Processes that are spawned by the Module are owned by the Module and are not owned by external processes/operators.

6.2 Configuration Settings and Restrictions

There are no security rules, settings, or restrictions to the configuration of the operational environment beyond the initialization instructions to set the module in Approved mode.

7 Physical Security

7.1 Mechanisms and Actions Required

The module is a multi-chip standalone hardware cryptographic module. The module meets the FIPS 140-3 Level 1 security requirements as production-grade components. The module has a multi-chip standalone embodiment and is made of commercially available, production-grade components meeting commercial specifications for power, temperature, reliability, shock and vibration. All production-grade components include standard passivation techniques, in the form of a coating applied over the module's circuitry to protect against environmental and other physical damage. The module is entirely contained within a production-grade metal enclosure.

8 Non-Invasive Security

Currently, the ISO/IEC 19790:2012 non-invasive security area is not required by FIPS 140-3 (see NIST SP 800-140F). The requirements of this area are not applicable to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

The table below lists the areas within the module's cryptographic boundary where SSPs can be stored.

Storage Area Name	Description	Persistence Type
RAM	Volatile Memory	Dynamic
Flash	Internal flash memory storage drive	Static

Table 13: Storage Areas

9.2 SSP Input-Output Methods

The table below lists the method used by the module for the input and output of SSPs.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Call Stack Input	RAM	External	Plaintext	Automated	Electronic	
API Call Stack output	External	RAM	Plaintext	Automated	Electronic	
SSP Input (Peer)	External	RAM	Plaintext	Automated	Electronic	
MACsec SSP Input	RAM	RAM	Encrypted	Automated	Electronic	
MACsec SSP Output	RAM	RAM	Encrypted	Automated	Electronic	
Manufacturer Installed	External	Flash	Plaintext	Manual	Electronic	

Table 14: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The table below describes the SSP zeroization methods employed by the module.

Zeroization Method	Description	Rationale	Operator Initiation
Zeroize CLI command	This command erases all data, including all configuration information, returning the module to its factory default state. The system is then rebooted	This command erases all the keys and CSPs from storage. The forced power cycle also zeroizes SSPs in volatile memory.	Invocation of the zeroize CLI command
Reset	Zeroization of SSPs in RAM via invocation of local or remoted reset service.	RAM is volatile and all data is lost when power is taken off. Zeroization is practically instantaneous.	Invocation of Local Reset or Remote Reset services.
Zeroize function	Zeroization of SSPs in memory when no longer needed.	Use of explicit zeroization function destroys SSP information immediately by overwriting memory area with zeroes	Operator cannot directly initiate this method.

Table 15: SSP Zeroization Methods

The Zeroize CLI command method is detailed in section 11.2.3.

The zeroization is initiated using a zeroization command or the zeroization function call, the completion of the command or function call indicates that zeroization has been completed. If the zeroization is initiated by power cycling the module, then a successful reboot of the module indicates that zeroization has completed successfully. In the case of zeroization initiated by session termination, SSPs are zeroized when the session terminates, and session termination is

indicated in the log. The design of the modules ensures that all data output, via the data output interface, is inhibited until the completion of the zeroization service.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC_DRBG V value	A critical value of the internal state of DRBG	256 bits - 256 bits	DRBG Internal State - CSP	DRBG (Kernel)		DRBG (Kernel)
HMAC DRBG key value	Critical value of the internal state of DRBG	256 bits - 256 bits	DRBG Internal State - CSP	DRBG (Kernel)		DRBG (Kernel)
HMAC_DRBG Entropy Input	A critical value of the internal state of DRBG provided by entropy source	256 bits - 256 bits	Entropy Source Output - CSP			DRBG (Kernel)
HMAC_DRBG Seed	Seed material used to the seed or reseed the HMAC DRBG	256 bits - 256 bits	DRBG Internal State - CSP	DRBG (Kernel)		DRBG (Kernel)
SSH DH Shared Secret	Shared DH value computed from the ephemeral DH key-pairs as part of SSH and used to derive session keys.	2048 bits - 112 bits	Shared Secret - CSP		Full KAS (SSH)	Full KAS (SSH)
SSH Private Host Key	Used for SSH host/server authentication	2048 bits - 112 bits	Asymmetric Private Key - CSP	KeyGen (SSH)		SigGen (SSH)
SSH Public Host Key	SSH Public Host Key	256, 384, 521, 2048, 4096 - 112, 128, 152, 192, 256	Asymmetric public key - PSP	KeyGen (SSH)		SigVer (SSH)
SSH DH Private Key	Used as part of DH key agreement protocol	FC, MODP-2048 - 112	Asymmetric Private Key - CSP	KAS KeyGen (SSH)		Full KAS (SSH)
SSH DH Public Key	Used as part of DH key agreement protocol	FC, MODP-2048 - 112	Asymmetric Public Key - PSP	KAS KeyGen (SSH)		Full KAS (SSH)
SSH-SEK	SSH Session Encryption Key	128, 192, 256 - 128, 192, 256 bits	Symmetric Key - CSP	KDF (SSH)		Symmetric Encryption / Decryption
Login Password	Used for authentication of the crypto officer and user	length range 10-20 characters - Please see Authentication Methods table for details.	Authentication password - CSP			SHA (libMD)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Root CA	X.509 Certificate used to verify the validity of the Juniper Package CA	256, 384 bits - 128, 192 bits	Asymmetric public key - CSP			Verify Image
Package CA	X.509 Certificate used to verify the validity of the Juniper Image at software load and also at runtime for integrity	256 bits - 128 bits	Asymmetric public key - PSP			Verify Image
MACsec CAK	Externally generated pre-shared key entered when MACsec static connectivity association key (CAK) security mode is enabled.	32 (hex) characters for 128-bit AES keys, 64 (hex) characters for 256-bit AES keys - 128 bits, 256 bits	Symmetric key - CSP			
MACsec CKN	Externally generated pre-shared key used to identify the CAK (64 characters)	64 characters -	Identifier - PSP			
MACsec PSK	MACsec PSK	128 bits, 256 bits - 128 bits, 256 bits	Pre-Shared Key - CSP			
MACsec SAK	Security Association Key used to encrypt/decrypt traffic for a given session	128 - 256 bits - 128 - 256 bits	Symmetric key - CSP	Key Derivation (MACsec)	Key wrap (MACsec)	Encrypt / Decrypt (MACsec)
MACsec KEK	Key Encryption Key used to transmit SAK to other members of a MACsec connectivity association	128 bits, 256 bits - 128 bits, 256 bits	Symmetric key - CSP	Key Derivation (MACsec)		Key wrap (MACsec)
MACsec ICK	Integrity Check Key used to verify the integrity and authenticity of the MPDUs	128 bits, 256 bits - 128 bits, 256 bits	Symmetric key - CSP	Key Derivation (MACsec)		Integrity (MACsec)
Operator RSA Public Key	Used for public key authentication of the crypto	2048 bits, 3072 bits, 4096 bits - 112	Authentication - PSP			

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	officer and user via SSH management session.	bits, 128 bits, 152 bits				
ECDH Shared Secret	Used as part of ECDH key agreement protocol	P-256, P-384, P-521 - 112 bits, 192 bits, 256 bits	Shared Secret - CSP		KAS-SSC (SSH)	KAS-SSC (SSH)
ECDH Public Key	Used as part of ECDH key agreement protocol	P-256, P-384, P-512 - 112 bits, 192 bits, 256 bits	Public Key - PSP	ECDSA KeyGen (FIPS186-5) (A6931)		KAS-SSC (SSH) KAS KeyGen (SSH) Full KAS (SSH)
ECDH Private Key	Used as part of ECDH key agreement protocol	P-256, P-384, P-512 - 112 bits, 192 bits, 256 bits	Private Key - CSP			KAS-SSC (SSH) Full KAS (SSH)
SSH Integrity Keys	Used for calculating MAC	224 bits, 256 bits, 512 bits - 224 bits, 256 bits, 512 bits	Symmetric Key - CSP	KDF (SSH)		MAC (SSH)

Table 16: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
HMAC_DRBG V value	API Call Stack Input API Call Stack output	RAM:Plaintext	Until updated by DRBG	Zeroize CLI command Reset Zeroize function	
HMAC DRBG key value	API Call Stack Input API Call Stack output	RAM:Plaintext	Until updated by DRBG	Zeroize CLI command Reset Zeroize function	
HMAC_DRBG Entropy Input	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	
HMAC_DRBG Seed	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	
SSH DH Shared Secret	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH DH Private Key:Computed SSH DH Public Key:Computed

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SSH Private Host Key	API Call Stack Input API Call Stack output	RAM:Plaintext Flash:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH Public Host Key:Paired With
SSH Public Host Key	API Call Stack Input SSP Input (Peer)	RAM:Plaintext Flash:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH Private Host Key:Paired With
SSH DH Private Key	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH DH Public Key:Paired With
SSH DH Public Key	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH DH Private Key:Paired With
SSH-SEK	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH DH Shared Secret:Derived From SSH Integrity Keys:Paired With
Login Password	API Call Stack Input API Call Stack output	RAM:Plaintext Flash:Encrypted	Until service completion	Zeroize CLI command Reset Zeroize function	
Root CA	Manufacturer Installed	RAM:Plaintext Flash:Plaintext	Until service completion	Zeroize CLI command	
Package CA	API Call Stack Input API Call Stack output Manufacturer Installed	RAM:Plaintext Flash:Plaintext	Until service completion	Zeroize CLI command	
MACsec CAK	API Call Stack Input API Call Stack output	RAM:Plaintext Flash:Obfuscated	Until service completion	Zeroize CLI command	MACsec CKN:Used With
MACsec CKN	MACsec SSP Input MACsec SSP Output	RAM:Plaintext Flash:Obfuscated	Until service completion	Zeroize CLI command Reset	MACsec CAK:Used With
MACsec PSK	MACsec SSP Input MACsec SSP Output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	MACsec ICK:Used With MACsec SAK:Used With MACsec KEK:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
MACsec SAK	MACsec SSP Input MACsec SSP Output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset	MACsec CAK:Derived From MACsec KEK:Used With
MACsec KEK	MACsec SSP Input MACsec SSP Output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset	MACsec CAK:Derived From MACsec SAK:Used With
MACsec ICK	API Call Stack Input API Call Stack output MACsec SSP Input MACsec SSP Output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	MACsec CAK:Derived From
Operator RSA Public Key	API Call Stack Input API Call Stack output	Flash:Plaintext		Zeroize CLI command	
ECDH Shared Secret	API Call Stack Input API Call Stack output	RAM:Plaintext		Zeroize CLI command	ECDH Private Key:Computed ECDH Public Key:Computed
ECDH Public Key	API Call Stack Input SSP Input (Peer)	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	ECDH Private Key:Paired With
ECDH Private Key	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	ECDH Public Key:Paired With
SSH Integrity Keys	API Call Stack Input API Call Stack output	RAM:Plaintext	Until service completion	Zeroize CLI command Reset Zeroize function	SSH DH Shared Secret:Derived From

Table 17: SSP Table 2

9.5 Transitions

In compliance with NIST SP 800-131A Rev. 2, the module supports algorithms and key lengths that provide a minimum of 112 bits of security strength for applying cryptographic protection. Starting January 1, 2031, the minimum-security strength for applying cryptographic protection will be 128 bits, and security strengths between 112 bits and 128 bits will be allowed for legacy use only to process information that is already protected. Please see the latest revision of SP 800-131A and CMVP Programmatic Transitions page for transitions that may affect this module.

Please note that the minimum key size transitions from 112 bits to 128 bits in 2030.

Please see the latest revision of SP 800-131 and CMVP Programmatic Transitions page for transitions that may affect this module.

10 Self-Tests

On power up or reset, the module performs the pre-operational self-tests and the indicated conditional cryptographic algorithm self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. The CASTs for algorithms utilized in the pre-operational firmware integrity check are performed prior to the firmware integrity check. The design of the modules ensures that all data output, via the data output interface, is inhibited whenever the module is in a pre-operational self-test condition.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
ECDSA SigVer / SHA2-256	P-256	Approved Integrity Technique	SW/FW Integrity	PASS/FAIL console output	Firmware Integrity Test

Table 18: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG (A6929)	256, SHA2-256	KAT	CAST	Pass Indicator / Error Indicator	Instantiate(), Reseed() and Generate()	Module Startup - On Demand
HMAC-SHA2-256 (A6929)	Key size : 256	KAT	CAST	Pass Indicator / Error Indicator	MAC	Module Startup - On Demand
SHA2-384 (A6929)		KAT	CAST	Pass Indicator / Error Indicator	HASH	Module Startup - On Demand
SHA2-512 (A6929)		KAT	CAST	Pass Indicator / Error Indicator	HASH	Module Startup - On Demand
SHA2-512 (A6930)		KAT	CAST	Pass Indicator / Error Indicator	HASH	Module Startup - On Demand
ECDSA	P-224	KAT	CAST	Pass Indicator / Error Indicator	ECDSA SigGen/SigVer	Module Startup - On Demand
HMAC-SHA2-224 (A6931)	Key size: 224	KAT	CAST	Pass Indicator / Error Indicator	MAC	Module Startup - On Demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A6931)	Key size: 256	KAT	CAST	Pass Indicator / Error Indicator	MAC	Module Startup - On Demand
HMAC-SHA2-512 (A6931)	Key size: 512	KAT	CAST	Pass Indicator / Error Indicator	MAC	Module Startup - On Demand
AES-CBC (A6931) Encrypt	Key size: 128	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Encryption	Module Startup - On Demand
AES-CBC (A6931) Decrypt	Key size: 128	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Decryption	Module Startup - On Demand
AES-GCM (A6931) Encrypt	Key size: 128	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Encryption	Module Startup - On Demand
AES-GCM (A6931) Decrypt	Key size: 128	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Decryption	Module Startup - On Demand
RSA SigVer	RSA 2048 w/ SHA2-256, RSA 4096 w/ SHA2-256	KAT	CAST	Pass Indicator / Error Indicator	Signature Verification	Module Startup - On Demand
RSA SigGen	RSA 2048 w/ SHA2-256, RSA 4096 w/ SHA2-256	KAT	CAST	Pass Indicator / Error Indicator	Signature Generation	Module Startup - On Demand
KDF SSH (A6931)	SHA2-256, SHA2-384	KAT	CAST	Pass Indicator / Error Indicator	Key Derivation	Module Startup - On Demand
KAS-ECC-SSC Sp800-56Ar3 (A6931)	P-256, P-384, P-521	KAT	CAST	Pass Indicator / Error Indicator	Shared Secret Computation	Module Startup - On Demand
KAS-FFC-SSC Sp800-56Ar3 (A6931)	MODP-2048, FC	KAT	CAST	Pass Indicator / Error Indicator	Shared Secret Computation	Module Startup - On Demand
AES-CMAC (A6928)	128 bits	KAT	CAST	Pass Indicator / Error Indicator	Message Authentication	Module Startup - On Demand
AES-ECB (A6928) Encrypt	128 bits	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Encryption	Module Startup - On Demand
AES-ECB (A6928) Decrypt	128 bits	KAT	CAST	Module is in operational state	Symmetric Decryption	Module Startup - On Demand
AES-KW (A6928) (Wrap)	128 bits	KAT	CAST	Pass Indicator / Error Indicator	Key Wrapping	Module Startup - On Demand
AES-KW (A6928) (Unwrap)	128 bits	KAT	CAST	Pass Indicator / Error Indicator	Key Unwrapping	Module Startup - On Demand
KBKDF	Key Derivation	KAT	CAST	Pass Indicator / Error Indicator	Key Derivation	Module Startup - On Demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA KeyGen (FIPS186-5) (A6931)	P-256	PCT	PCT	Pass Indicator / Error Indicator	ECDSA KeyGen KeyVer PCT	On Key/Signature Generation
KAS-FFC-SSC Sp800-56Ar3	MODP-2048	PCT	PCT	Pass Indicator / Error Indicator	DH KeyGen KeyVer PCT	On Key Generation
HMAC-SHA2-256 (A6930)	Key size: 256	KAT	CAST	Pass Indicator / Error Indicator	Message Authentication	Module Startup - On Demand
DSA KeyGen (FIPS186-4) (A6931)	DSA KeyGen	PCT	PCT	Pass Indicator / Error Indicator	DSA KeyVer	On Key Generation
AES-GCM (MACSec HW) Encrypt	Key size: 256	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Encryption	Module Startup - On Demand
AES-GCM (MACSec HW) Decrypt	Key size: 256	KAT	CAST	Pass Indicator / Error Indicator	Symmetric Decryption	Module Startup - On Demand
Firmware Load Test	ECDSA P-256 with SHA2-256	KAT	SW/FW Load	Pass Indicator / Error Indicator	Verification of ECDSA signature on FW	On firmware load
RSA KeyGen (FIPS186-5) (A6931)	RSA KeyGen	PCT	PCT	Pass Indicator / Error Indicator	RSA Key Generation	Module Startup - On Demand
Entropy Source - RCT	-	RCT	CAST	Module is in operational state	-	On power-up
Entropy Source - APT	-	APT	CAST	Module is in operational state	-	On power-up

Table 19: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer / SHA2-256	Approved Integrity Technique	SW/FW Integrity	Module startup	Programmatically

Table 20: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A6929)	KAT	CAST	On Startup	Manually power cycle host or programmatically
HMAC-SHA2-256 (A6929)	KAT	CAST	On Startup	Manually power cycle host or programmatically

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-384 (A6929)	KAT	CAST	On Startup	Manually power cycle host or programmatically
SHA2-512 (A6929)	KAT	CAST	On Startup	Manually power cycle host or programmatically
SHA2-512 (A6930)	KAT	CAST	On Startup	Manually power cycle host or programmatically
ECDSA	KAT	CAST	On Startup	Manually power cycle host or programmatically
HMAC-SHA2-224 (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
HMAC-SHA2-256 (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
HMAC-SHA2-512 (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-CBC (A6931) Encrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-CBC (A6931) Decrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-GCM (A6931) Encrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-GCM (A6931) Decrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
RSA SigVer	KAT	CAST	On Startup	Manually power cycle host or programmatically
RSA SigGen	KAT	CAST	On Startup	Manually power cycle host or programmatically
KDF SSH (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
KAS-ECC-SSC Sp800-56Ar3 (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
KAS-FFC-SSC Sp800-56Ar3 (A6931)	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-CMAC (A6928)	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-ECB (A6928) Encrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A6928) Decrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-KW (A6928) (Wrap)	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-KW (A6928) (Unwrap)	KAT	CAST	On Startup	Manually power cycle host or programmatically
KBKDF	KAT	CAST	On Startup	Manually power cycle host or programmatically
ECDSA KeyGen (FIPS186-5) (A6931)	PCT	PCT	On Startup	Manually power cycle host or programmatically
KAS-FFC-SSC Sp800-56Ar3	PCT	PCT	On Startup	Manually power cycle host or programmatically
HMAC-SHA2-256 (A6930)	KAT	CAST	On Startup	Manually power cycle host or programmatically
DSA KeyGen (FIPS186-4) (A6931)	PCT	PCT	On Startup	Manually power cycle host or programmatically
AES-GCM (MACSec HW) Encrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
AES-GCM (MACSec HW) Decrypt	KAT	CAST	On Startup	Manually power cycle host or programmatically
Firmware Load Test	KAT	SW/FW Load	On firmware load	Automatic
RSA KeyGen (FIPS186-5) (A6931)	PCT	PCT	On Startup	On key generation
Entropy Source - RCT	RCT	CAST	On Startup	Manually
Entropy Source - APT	APT	CAST	On Startup	Manually

Table 21: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Critical Failure State	The cryptographic module ceases to perform cryptographic operations, inhibits all data output, and provides status of the error via syslog messages and console status output	On any power-up self-test or PCT failure	Power cycle	Console status indicator
Soft Error State	A non-critical self-test failure occurs, causing a failure of the triggering operation	Firmware load test or continuous	The module processes the error,	Console displays error

Name	Description	Conditions	Recovery Method	Indicator
		entropy health test failure	and resumes normal operation	

Table 22: Error States

The module enters error state upon failure of any self-tests, causing the kernel to 'panic' and all execution to halt. The only way to exit from this state is to reboot the module, which causes the self-tests to be repeated and pass successfully before the corresponding algorithms are usable.

10.5 Operator Initiation of Self-Tests

Self–tests that are performed at power-up are available on demand by power cycling the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module must be correctly installed and configured to enter a FIPS compliant state and operate in the Approved mode. The required procedures are as follows:

1. Install the Junos OS firmware image - the procedure is detailed in Section 11.2.1
2. Configure device for the Approved mode - the procedure is Section 11.2.2.

To continue using the module in a FIPS compliant way, the Module Operation Rules in Section 11.4.2 must be followed.

11.2 Administrator Guidance

11.2.1Installing the Junos OS firmware image

1. 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.
   The cryptographic module devices use the following firmware image:
   EX4400 series - 24.2R2-S3.3
2. Connect to the console port on the device from your management device, and log in to the Junos OS CLI.
3. Install the new package on the device (package may be a local file copied to the device, or a file on a remote server):
   ```
   user@host> request system software add <package>
   ```

4. Reboot the device to load the installation:
   ```
   user@host> request system reboot
   ```
5. After the reboot has completed, log in and use the show version command to verify that the new version of the software is successfully installed.
   ```
   user@host> show version
   ```

11.2.2 Configure the device for the Approved mode

To configure the device for the Approved mode:

1. Zeroize the device to delete all CSPs before entering FIPS mode.
   ```
   root@host# request system zeroize
   ```
2. After the device comes up, login using username “root” and password blank.
3. Configure root authentication with password at least 10 characters or more.
   ```
   root@host# set system root-authentication plain-text-password
   ```
4. Load configuration onto device and commit new configuration.

   NOTE: SSH key-exchange configuration must not include ‘dh-group14-sha1’. It is not approved for this module.
5. Configure Crypto Officer and login with Crypto Officer credentials.
6. Set the fips level to 1.
   ```
   request system software add optional://fips-mode.tgz
   crypto-officer@host# set system fips level 1
   ```
7. Commit and reboot the device.
   ```
   crypto-officer@host# commit
   crypto-officer@host# run request system reboot
   ```

11.2.3 Administrator Module Operational Rules

1. The Crypto Officer must retain control of the module while zeroization is in process.
2. The Crypto Officer shall verify that the firmware image to be loaded on the module is a FIPS validated image.
3. Before pushing the factory reset button on the device, the cryptographic officer shall perform the zeroize command as described in Section 11.2.3.
4. The password minimum-length must be configured to be at least 10.
5. Virtual Chassis features must not be configured.
6. Dynamic CAK mode shall not be configured for MACsec.
7. Only the AES-GCM cipher suites shall be configured for MACsec.
8. The module shall only be used with CMVP-validated modules when supporting the MACsec protocol for providing Peer, Authenticator functionality.
9. The link between the Peer and Authenticator, used in the MACsec communication, shall be secure to prevent the possibility for an attacker to introduce foreign equipment into the local area network.
10. The module shall not be configured to use a RADIUS server and the RADIUS server capability shall be disabled.
11. SSH key-exchange must not be configured to include ‘dh-group14-sha1’.

11.2.4 Zeroizing the System

To zeroize the device:

1. Login to the device as Crypto Officer and from CLI, enter
```
crypto-officer@host# request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes, no] (no)
```

2. To initiate the zeroization process, type yes at the prompt:
```
Erase all data, including configuration and log files? [yes, no] (no)
yes
```

11.3 Non-Administrator Guidance

No specific non-administrator guidance is required to operate the module.

11.4 Design and Rules

11.4.1 Module Design Rules

The module design implements the following security rules:

1. The module clears previous authentications on power cycle.
2. Power up self-tests do not require any operator action.
3. Data output is inhibited during key generation, self-tests, zeroization, and error states.
4. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
5. There are no restrictions on which SSPs are zeroized by the zeroization service.
6. The module does not support a maintenance interface or role.
7. The module requires two independent internal actions to be performed prior to outputting plaintext CSPs.

11.5 End of Life

When disposing of the cryptographic module, the Crypto Officer shall perform the zeroize command as described in Section 11.2.3.
CAUTION: Perform system zeroization with care. After the zeroization process is complete, no data is left on the device. The device is returned to the factory default state, equivalent to a fresh installation of the firmware, without any configured users or configuration files.

After zeroizing the system, the module is no longer in a FIPS compliant state. (Installation and configuration as per Section 11.1 is required to enter the FIPS compliant state and enable the Approved mode of operation).

NOTE: The Crypto Officer must retain control of the module while zeroization is in progress.

12 Mitigation of Other Attacks

The module does not implement mechanisms to mitigate other attacks beyond what is described in this Security Policy.