# Juniper Networks MX240, MX480, MX960, MX2010, MX2020 3D Universal Edge Routers and EX9204, EX9208, EX9214 Ethernet Switches with RE-S-X6-64G/REMX2K-X8-64G/EX9200-RE2 Routing Engine

**Firmware: Junos OS 18.1R1**

# Non-Proprietary FIPS 140-2 Cryptographic Module Security Policy

**Version: 1.1**

**Date: December 20, 2018**

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408.745.2000
1.888 JUNIPER
www.juniper.net

# Contents



# List of Tables

## List of Figures

# 1 Introduction

This is a non-proprietary Cryptographic Module Security Policy for the Juniper Networks MX Series 3D Universal Edge Routers with the Next Generation Routing Engine (NGRE) routing engine (the "MX Series") and the EX9200 series switches with the NGRE routing engine. The MX series provides dedicated high-performance processing for flows and sessions and integrates advanced security capabilities that protect the network infrastructure as well as user data. The EX9200 series enables collaboration and provides simple and secure access for the delivery of mission-critical applications in the enterprise campus. In the data center, it simplifies operations to align the network with fast-changing business requirements.

This FIPS 140-2 validation includes the following MX series router models: the MX240, MX480, MX960, MX2010 and MX2020 and the following EX series switch models: EX9204, EX9208, EX9214. The FIPS validated version of firmware is Junos OS 18.1R1.

The cryptographic boundary for this MX Series and EX Series is defined as follows for the validation:

- the outer edge of the chassis includes the Routing Engine (RE), Switch Control Board (SCB/SFB/SF), slot cover in the following configurations:
  - For MX240 (2 available RE slots, 2 additional slots): 1 SCB and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - MX480 (2 available RE slots, 6 additional slots): 1 SCB and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For MX960 (2 available RE slots, 12 additional slots): 1 SCB and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For MX2010 (2 available RE slots, 10 additional slots): 1 SFB and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For MX2020 (2 available RE slots, 20 additional slots): 1 SFB and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For EX9204 (2 available RE slots, 2 additional slots): 1 SF module and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For EX9208 (2 available RE slots, 6 additional slots): 1 SF module and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
  - For EX9214 (2 available RE slots, 12 additional slots): 1 SF module and 1 RE. All empty module bays must have a slot cover installed for proper cooling air circulation.
- includes the inverse three-dimensional space where non-crypto-relevant line cards fit, with the backplane port serving as the physical interface
- excluding the power distribution module on the rear of the device.

The cryptographic modules provide for an encrypted connection, using SSH, between the management station and the module. All other data input or output from the module is considered plaintext for this FIPS 140-2 validation.

The cryptographic modules are defined as multiple-chip standalone modules that execute Junos OS 18.1R1 firmware on any of the Juniper Networks MX 3D Universal Edge Routers listed in Table 1 below.

**Table 1 – Cryptographic Module Hardware Configurations**

| Chassis PN | Power PN | SCB PN | RE PN |
|---|---|---|---|
| MX240 | PWR-MX480-2400-DC<br>PWR-MX480-2520-AC | SCBE2-MX | RE-S-X6-64G |
| MX480 | PWR-MX480-2400-DC<br>PWR-MX480-2520-AC | SCBE2-MX | RE-S-X6-64G |
| MX960 | PWR-MX960-4100-DC<br>PWR-MX960-DC<br>PWR-MX960-4100-AC<br>PWR-MX960-AC-S | SCBE2-MX | RE-S-X6-64G |
| MX2010 | MX2K-PDM-OP-DC<br>MX2000-PDM-DC<br>MX2K-PDM-AC-1PH<br>MX2K-PDM-OP-AC | MX2K-SFB | REMX2K-X8-64G |
| MX2020 | MX2K-PDM-OP-DC<br>MX2000-PDM-DC<br>MX2K-PDM-AC-1PH<br>MX2K-PDM-OP-AC | MX2K-SFB | REMX2K-X8-64G |
| EX9204 | PWR-MX480-2400-DC<br>PWR-MX480-2520-AC | EX9200-SF2 | EX9200-RE2 |
| EX9208 | PWR-MX480-2400-DC<br>PWR-MX480-2520-AC | EX9200-SF2 | EX9200-RE2 |
| EX9214 | PWR-MX960-4100-DC<br>PWR-MX960-4100-AC | EX9200-SF2 | EX9200-RE2 |

Juniper also offers an enhanced routing engine that can be used in the MX series routers (RE-S-X6-128G, REMX2K-X8-128G). The enhanced routing engine offers SSDs with more storage capacity (2x100G), and RAM with more memory (128G) than the routing engines tested.

Juniper affirms that the enhanced routing engines use the same cryptographic functions that are available in the routing engines listed in Table 1. However, the RE-S-X6-128G and REMX2K-X8-128G routing engines were not tested for this FIPS 140-2 validation. No claim can be made as to the conformance of the MX series routers with the enhanced routing engine for they were not tested by a CSTL or reviewed by the CMVP.

The modules are designed to meet FIPS 140-2 Level 1 overall:

**Table 2- Security Level of Security Requirements**

| Area | Description | Level |
|---|---|---|
| 1 | Module Specification | 1 |
| 2 | Ports and Interfaces | 1 |
| 3 | Roles, Services, and Authentication | 3 |
| 4 | Finite State Model | 1 |
| 5 | Physical Security | 1 |
| 6 | Operational Environment | N/A |
| 7 | Key Management | 1 |
| 8 | EMI/EMC | 1 |
| 9 | Self-test | 1 |
| 10 | Design Assurance | 3 |
| 11 | Mitigation of Other Attacks | N/A |
| | *Overall* | 1 |

The modules have a limited operational environment as per the FIPS 140-2 definitions. It includes a firmware load service to support necessary updates. New firmware versions within the scope of this validation must be validated through the FIPS 140-2 CMVP. Any other firmware loaded into the modules are out of the scope of this validation and require a separate FIPS 140-2 validation.

The modules do not implement any mitigations of other attacks as defined by FIPS 140-2.

## 1.1 Hardware and Physical Cryptographic Boundary

The cryptographic modules' operational environment is a limited operational environment.

The image below depicts the physical boundary of the modules, including the Routing Engine and SCB. The boundary excludes the non-crypto-relevant line cards included in the figure.

**Figure 1 – Physical Cryptographic Boundary (Left to Right: MX240, MX480, MX960, MX2010, MX2020)**

**Figure 2 - Physical Cryptographic Boundary (Left to Right: EX9204, EX9208, EX9214)**

**Table 3 - Ports and Interfaces**

| Port | Description | Logical Interface Type |
|---|---|---|
| Ethernet (data) | LAN Communications | Control in, Data in, Status out, Data out |
| Ethernet (mgmt.) | Remote Management | Control in, Data in, Status out, Data out |
| Serial | Console serial port | Control in, Data in, Status out, Data out |
| Power | Power connector | Power |
| Reset Button | Reset | Control in |
| LED | Status indicator lighting | Status out |
| USB | Load Junos OS image | Control in, Data in |
| Backplane | Line card backplane interfaces | Control in, Data in, Status out, Data out |
| Chassis Cluster Control | Disabled | N/A |
| Aux | Disabled | N/A |

## 1.2 Modes of Operation

The module supports a FIPS Approved mode of operation and a non-Approved mode of operation. The module must always be zeroized when switching between a FIPS Approved mode of operation and the non-Approved mode of operation and vice versa.

### 1.2.1 FIPS Approved Mode

The Crypto-Officer places the module in an Approved mode of operation by following the instructions in the crypto-officer guidance (section 6.1).

The Crypto-Officer can verify that the cryptographic module is in an Approved mode by observing the console prompt and running the “show version” command. When operating in FIPS mode, the prompt will read “<user>@<device name>:fips#” (e.g. crypto-officer@mx240:fips#). The Crypto-Officer can also use the “show system fips level” command to determine if the module is operating in FIPS mode.

### 1.2.2 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 the algorithms identified in Section 2.2 as well as the algorithms supported in the Approved mode of operation.

The Crypto-Officer can place the module into a non-approved mode of operation by following the instructions in the crypto-officer guidance (section 6.1)

## 1.3 Zeroization

The cryptographic module provides a non-Approved mode of operation in which non-approved

cryptographic algorithms are supported. When transitioning between the non-Approved mode of operation and the Approved mode of operation, the Cryptographic Officer must run the following commands to zeroize the Approved mode CSPs:

```
co@device> request vmhost zeroize no-forwarding
```

This command wipes clean all the CSPs and configurations and then reboots the device and sets it to the factory-default configuration.

Use of the zeroize command is restricted to the Cryptographic Officer. The cryptographic officer shall perform zeroization in the following situations:

1. Before FIPS Operation: To prepare the device for operation as a FIPS cryptographic module by erasing all CSPs and other user-created data on a device before its operation as a FIPS cryptographic module.
2. Before non-FIPS Operation: To conduct erasure of all CSPs and other user-created data on a device in preparation for repurposing the device for non-FIPS operation.

> Note: The Cryptographic Officer must retain control of the module while zeroization is in process.

# 2 Cryptographic Functionality

## 2.1 Allowed Algorithms and Protocols

The module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in Tables 4, 5, 6, 7, 8 and 9 below. Table 10 summarizes the high-level protocol algorithm support.

**Table 4 – Kernel Approved Cryptographic Functions**

| CAVP Cert. | Algorithm | Standard | Mode | Description | Functions |
|---|---|---|---|---|---|
| 2146 | DRBG | SP800-90A | HMAC | SHA-256 | Random Bit Generation |
| 3623 | HMAC | PUB 198 | SHA-1 | Key size: 160 bits, λ = 96 | Message Authentication, DRBG Primitive |
| | | | SHA-256 | Key size: 256 bits, λ = 128, 256 | |
| 4386 | SHS | PUB 180-4 | SHA-1<br>SHA-256<br>SHA-384<br>SHA-512 | | Message Digest Generation |

**Table 5 – LibMD Approved Cryptographic Functions**

| CAVP Cert. | Algorithm | Standard | Mode | Description | Functions |
|---|---|---|---|---|---|
| 3622 | HMAC | PUB 198 | SHA-1 | Key size: 160 bits, λ = 96 | Message Authentication |
| | | | SHA-256 | Key size: 256 bits, λ = 128, 256 | |
| 4385 | SHS | PUB 180-4 | SHA-1<br>SHA-256<br>SHA-512 | | Message Digest Generation |

**Table 6 – OpenSSL Approved Cryptographic Functions**

| CAVP Cert. | Algorithm | Standard | Mode | Description | Functions |
|---|---|---|---|---|---|
| 5466[1] | AES | PUB 197-38A | CBC, ECB, CTR | Key Sizes: 128, 192, 256 | Encrypt, Decrypt |

[1] AES GCM, in the OpenSSL implementation, was validated by CAVP but is not used for any services in the Approved modes of operation.

| | | | | | |
|---|---|---|---|---|---|
| N/A[2] | CKG | SSH-PUB 133 | Section 6.1<br>Section 6.2 | | Asymmetric key generation using unmodified DRBG output |
| 1917 | CVL (KAS) | SP 800-56A | ECC DH | P-256 (SHA 256)<br>P-384 (SHA 384)<br>P-521 (SHA 512) | Key Agreement Scheme |
| 1918 | CVL | SP 800-135 | SSH | SHA 1, 256, 384, 512 | Key Derivation |
| 2147 | DRBG | SP 800-90A | HMAC | | SHA-256 |
| 1462 | ECDSA | PUB 186-4 | | P-256 (SHA 256)<br>P-384 (SHA 384)<br>P-521 (SHA 512) | SigGen, KeyGen, SigVer |
| 3624 | HMAC | PUB 198 | SHA-1 | Key size: 160 bits, λ = 160 | Message Authentication |
| | | | SHA-224 | Key size: 224 bits, λ = 192 | |
| | | | SHA-512 | Key size: 512 bits, λ = 512 | |
| | | | SHA-256 | Key size: 256, λ = 256 | Message Authentication DRBG Primitive |
| N/A | KTS | | AES Cert. #5466 and HMAC Cert. #3624 | | Key establishment methodology provides between 128 and 256 bits of encryption strength |
| | | | Triple-DES Cert. #2751 and HMAC Cert. #3624 | | key establishment methodology provides 112 bits of encryption strength |
| 2936 | RSA | PUB 186-4 | | n=2048 (SHA 256, 512)<br>n=3072 (SHA 256, 512)<br>n=4096 (SHA 256, 512) | KeyGen, SigGen, SigVer[3] |
| 4387 | SHS | PUB 180-4 | SHA-1<br>SHA-256<br>SHA-384<br>SHA-512 | | Message Digest Generation,<br>KDF Primitive |
| | | | SHA-224 | | Message Digest Generation |

[2] Vendor Affirmed

[3] RSA 4096 SigVer was not tested by the CAVP; however, it is Approved for use per CMVP guidance, because RSA 2048 SigVer was tested and testing for RSA 4096 SigVer is not available.

| | | | | | |
|---|---|---|---|---|---|
| 2751 | Triple-DES | SP 800-67 | TCBC | Key Size: 192 | Encrypt, Decrypt |

**Table 7 - Allowed Cryptographic Functions**

| Algorithm | Caveat | Use |
|---|---|---|
| Elliptic Curve Diffie-Hellman [IG] D.8 | Provides between 128 and 256 bits of encryption strength. | key agreement; key establishment |
| NDRNG [IG] 7.14 Scenario 1a | The module generates a minimum of 256 bits of entropy for key generation. | Seeding the DRBG |

**Table 8 - Protocols Allowed in FIPS Mode**

| Protocol | Key Exchange | Auth | Cipher | Integrity |
|---|---|---|---|---|
| SSHv2[4] | EC Diffie-Hellman P-256, P-384, P-521 | RSA 2048<br>ECDSA P-256 | Triple-DES CBC<br>AES CBC 128/192/256<br>AES CTR 128/192/256 | HMAC-SHA-1<br>HMAC-SHA-256<br>HMAC-SHA-512 |

No part of these protocols, other than the KDF, have been tested by the CAVP and CMVP. The SSH algorithms allow independent selection of key exchange, authentication, cipher and integrity. In Table 8 above, each column of options for a given protocol is independent and may be used in any viable combination.

## 2.2 Disallowed Algorithms and Protocols

These algorithms and protocols are non-Approved algorithms and protocols that are disabled when the module is operated in an Approved mode of operation. The algorithms are available as part of the SSH connect service when the module is operated in the non-Approved mode.

**Algorithms**

- RSA with key size less than 2048
- ECDSA with ed25519 curve
- ECDH with ed25519 curve
- AES-GCM
- ARCFOUR
- Blowfish
- CAST
- DSA (SigGen, SigVer; non-compliant)

[4] RFC 4253 governs the generation of the Triple-DES encryption key for use with the SSHv2 protocol

- HMAC-MD5
- HMAC-RIPEMD160
- UMAC

**Protocols**

- Finger
- ftp
- rlogin
- telnet
- tftp
- xnm-clear-text

## 2.3 Critical Security Parameters

All CSPs and public keys used by the module are described in this section.

**Table 9 - Critical Security Parameters (CSPs)**

| Name | Description and usage |
|---|---|
| DRBG_Seed | Seed material used to seed or reseed the DRBG |
| DRBG_State | Values V and Key which comprise the HMAC_DRBG state |
| Entropy Input | 256 bits entropy (min) input used to instantiate the DRBG |
| SSH PHK | SSH Private host key. 1st time SSH is configured, the keys are generated. ECDSA P-256. RSA2048 |
| SSH ECDH | Ephemeral EC Diffie-Hellman private key used in SSH. ECDH P-256, P-384, or P-521 |
| SSH-SEKs | SSH Session Keys: SSH Session Encryption Key: 3-Key Triple-DES or AES (128,192,256); SSH Session Integrity Key: HMAC. |
| User Password | Passwords used to authenticate Users to the module. |
| CO Password | Passwords used to authenticate COs to the module. |

**Table 10 - Public Keys**

| Name | escription and usage |
|---|---|
| SSH-PUB | SSH Public Host Key used to identify the host. ECDSA P-256. RSA 2048 |
| SSH-ECDH-PUB | Ephemeral EC Diffie-Hellman public key used in SSH key establishment. ECDH P-256, P-384, or P-521 |
| Auth-User Pub | User Authentication Public Keys. Used to authenticate users to the module. ECDSA P-256, P-384, or P-521 |
| Auth-CO Pub | CO Authentication Public Keys. Used to authenticate CO to the module. ECDSA P-256, P-384, or P-521 |
| Root CA | ECDSA P-256 X.509 Certificate; Used to verify the validity of the Juniper Package CA at software load and also at runtime for integrity. |
| Package CA | ECDSA P-256 X.509 Certificate; Used to verify the validity the Juniper Image at software load and also at runtime for integrity. |

# 3 Roles, Authentication and Services

## 3.1 Roles and Authentication of Operators to Roles

The module supports two roles: Cryptographic Officer (CO) and User. The module supports concurrent operators, but does not support a maintenance role and/or bypass capability. The module enforces the separation of roles using identity-based operator authentication.

The Cryptographic Officer role configures and monitors the module via a console or SSH connection. As root or super-user, the Cryptographic Officer has permission to view and edit secrets within the module.

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

## 3.2 Authentication Methods

The module implements two forms of Identity-Based authentication, Username and password over the Console and SSH as well as Username and ECDSA or RSA public key over SSH.

Password authentication: 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 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. $4^{th}$ failed attempt = 10-second delay, $5^{th}$ failed attempt = 15-second delay, $6^{th}$ failed attempt = 20-second delay, $7^{th}$ 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 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.

ECDSA signature verification: SSH public-key authentication. Processing constraints allow for a maximum of 5.6e7 ECDSA attempts per minute. The module supports ECDSA (P-256, P-384, and P-521), which has a minimum equivalent computational resistance to attack of either $2^{128}$ depending on the curve. Thus the probability of a successful random attempt is $1/ (2^{128})$, which is less than 1/1,000,000. Processing speed (partial establishment of an SSH session) limits the number of failed authentication attempts in a one-minute period to 5.6e7 attempts. The probability of a success with multiple consecutive attempts in a one-minute period is $5.6e7/(2^{128})$, which is less than 1/100,000.

RSA signature verification: SSH public-key authentication. Processing constraints allow for a maximum of 5.6e7 RSA attempts per minute. The module supports RSA (2048, 4096), which has a minimum equivalent computational resistance to attack of $2^{112}$ (2048). Thus, the probability of a successful

random attempt is 1/ ($2^{112}$), which is less than 1/1,000,000. Processing speed (partial establishment of an SSH session) limits the number of failed authentication attempts in a one-minute period to 5.6e7 attempts. The probability of a success with multiple consecutive attempts in a one-minute period is 5.6e7/ ($2^{112}$), which is less than 1/100,000.

### 3.3 Services

All services implemented by the module are listed in the tables below. Table 13 lists the access to CSPs by each service.

**Table 11 - Authenticated Services**

| Service | Description | CO | User |
|---|---|---|---|
| Configure security | Security relevant configuration | x | |
| Configure | Non-security relevant configuration | x | |
| Status | Show status | x | x |
| Zeroize | Destroy all CSPs | x | |
| SSH connect | Initiate SSH connection for SSH monitoring and control (CLI) | x | x |
| Console access | Console monitoring and control (CLI) | x | x |
| Remote reset | Software initiated reset, Performs self-tests on demand. | x | |
| Load Image | Verification and loading of a validated firmware image | x | |

**Table 12 - Unauthenticated Services**

| Service | Description |
|---|---|
| Local reset | Hardware reset or power cycle |
| Traffic | Traffic requiring no cryptographic services (e.g. OSPF, BGP) |
| LED Status | Basic |

**Table 13 - CSP Access Rights within Services**

| Service | CSPs | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | DRBG_Seed | DRBG_State | Entropy Input String | SSH PHK | SSH ECDH | SSH-SEK | CO-PW | User-PW |
| Configure security | -- | E | -- | GWR | -- | -- | W | W |
| Configure | -- | -- | -- | -- | -- | -- | -- | -- |
| Secure traffic | -- | -- | -- | -- | -- | -- | -- | -- |
| Status | -- | -- | -- | -- | -- | -- | -- | -- |
| Zeroize | Z | Z | Z | Z | Z | Z | Z | Z |
| SSH connect | -- | E | -- | E | GE | GE | E | E |
| Console access | -- | -- | -- | -- | -- | -- | E | E |
| Remote reset | GEZ | GZ | GZ | -- | Z | Z | -- | -- |
| Load Image | -- | -- | -- | -- | -- | -- | -- | -- |
| Local reset | GEZ | GZ | GZ | -- | Z | Z | -- | -- |
| Traffic | -- | -- | -- | -- | -- | -- | -- | -- |

G = Generate: The module generates the CSP
R = Read: The CSP is read from the module (e.g. the CSP is output)
E = Execute: The module executes using the CSP
W = Write: The CSP is updated or written to the module (persistent storage) Z = Zeroize: The module zeroizes the CSP.

## 3.4 Non-Approved Services

The following services are available in the non-Approved mode of operation. The security functions provided by the non-Approved services are identical to the Approved counterparts with the exception of SSH Connect (non-compliant). SSH Connect (non-compliant) supports the security functions identified in Section 2.2 and the SSHv2 row of Table 8.

**Table 14 - - Authenticated Services**

| Service | Description | CO | User |
|---|---|---|---|
| Configure security (non-compliant) | Security relevant configuration | x | |
| Configure | Non-security relevant configuration | x | |
| Status (non-compliant) | Show status | x | x |
| Zeroize (non-compliant) | Destroy all CSPs | x | |
| SSH connect (non-compliant) | Initiate SSH connection for SSH monitoring and control (CLI) | x | x |
| Console access (non-compliant) | Console monitoring and control (CLI) | x | x |
| Remote reset (non-compliant) | Software initiated reset. Performs self-tests on demand. | x | |
| Load Image (non-compliant) | Verification and loading of a validated firmware image into the device. | x | |

**Table 15 - Unauthenticated Services**

| Service | Description |
|---|---|
| Local reset | Hardware reset or power cycle |
| Traffic | Traffic requiring no cryptographic services (e.g. OSPF, BGP) |
| LED Status | Basic |

# 4 Self-tests

Each time the module is powered up it tests that the cryptographic algorithms still operate correctly, and that sensitive data have not been damaged. Power-up self–tests are available on demand by power cycling the module (Remote reset service).

On power up or reset, the module performs the self-tests described below. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fails, the module enters the Critical Failure error state.

The module performs the following power-up self-tests:

- Firmware Integrity check using ECDSA P-256 with SHA-256
- **Kernel KATs**
  - AES-CBC (128/192/256) Encrypt KAT
  - AES-CBC (128/192/256) Decrypt KAT
  - SP 800-90A HMAC DRBG KAT
    - Health-tests initialize, re-seed, and generate
  - HMAC-SHA-1 KAT
  - HMAC-SHA-256 KAT
  - SHA-384 KAT
  - SHA-512 KAT
  - Triple-DES-CBC Encrypt KAT
  - Triple-DES-CBC Decrypt KAT
- **OpenSSL KATs**
  - AES-CBC (128/192/256) Encrypt KAT
  - AES-CBC (128/192/256) Decrypt KAT
  - SP 800-90A HMAC DRBG KAT
    - Health-tests initialize, re-seed, and generate
  - ECDSA P-256 Sign/Verify PCT
  - ECDH P-256 KAT
    - Derivation of the expected shared secret.
  - HMAC-SHA-1 KAT
  - HMAC-SHA-224 KAT
  - HMAC-SHA-256 KAT
  - HMAC-SHA-512 KAT
  - KAS -ECC
  - KDF-SSH KAT
  - RSA 2048 w/ SHA-256 Sign KAT
  - RSA 2048 w/ SHA-256 Verify KAT
  - SHA-384 KAT
  - Triple-DES-CBC Encrypt KAT
  - Triple-DES-CBC Decrypt KAT
- **LibMD KATs**
  - HMAC SHA-1
  - HMAC SHA-256
  - SHA-512

- Critical Function Test
    - The cryptographic module performs a verification of a limited operational environment, and verification of optional non-critical packages.

The module also performs the following conditional self-tests:

- Continuous RNG Test on the OpenSSL SP 800-90A HMAC-DRBG
- Continuous RNG test on the NDRNG
- Pairwise consistency test when generating ECDSA, and RSA key pairs.
- Firmware Load Test (ECDSA signature verification)

# 5 Physical Security Policy

The modules physical embodiment is that of a multi-chip standalone device that meets Level 1 Physical Security requirements. The module is completely enclosed in a rectangular nickel or clear zinc coated, cold rolled steel, plated steel and brushed aluminum enclosure.

# 6 Security Rules and Guidance

The module design corresponds to the security rules below. The term *must* 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. Power up self-tests do not require any operator action.
4. Data output is inhibited during key generation, self-tests, zeroization, and error states.
5. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
6. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
7. The module does not support a maintenance interface or role.
8. The module does not support manual key entry.
9. The module does not output intermediate key values.
10. The module requires two independent internal actions to be performed prior to outputting plaintext CSPs.
11. The cryptographic officer must determine whether firmware being loaded is a legacy use of the firmware load service.
12. The cryptographic officer must retain control of the module while zeroization is in process.
13. The Triple-DES encryption key is generated as part of recognized IETF protocols (RFC RFC 4253 SSH). The operator shall ensure that the number of 64-bit blocks encrypted by the same key does not exceed 2^20.
14. Virtual Chassis is not supported in FIPS mode and shall not be configured on the modules.
15. RSA key generated shall only be 2048 bits or greater.

## 6.1 Crypto-Officer Guidance

### 6.1.1 Enabling FIPS-Approved Mode of Operation

The crypto-officer is responsible for initializing the module in a FIPS Approved mode of operation. The FIPS-Approved mode of operation is not automatically enabled. The Crypto-officer should execute the following steps:

1. Zeroize the device according to the instructions in the section 1.3.

2. To enable FIPS mode in Junos OS on the device:

   a. Enter configuration mode:

```
co@deivce> configure
Entering configuration mode
[edit]
co@device#
```

   b. Enable FIPS mode on the device by setting the FIPS level to 1, and verify the level:

```
[edit]
co@device# set system fips level 1
[edit]
co@device# show system fips level
level 1;
```

   c. Commit the configuration:

```
[edit]
co@device# commit
configuration check succeeds
[edit]
  'system'
    reboot is required to transition to FIPS level 1
commit complete
```

   d. Reboot the device:

```
[edit]
co@device# run request system reboot
Reboot the system ? [yes,no] (no) yes
```

### 6.1.2 Placing the Module in a Non-Approved Mode of Operation

As cryptographic officer, the operator need to disable the FIPS-Approved mode of operation on the device to return it to a non-Approved mode of operation. To disable FIPS-Approved mode, the module must be zeroized. Follow the steps found in section 1.3 to zeroize the module.

## 6.2 User Guidance

The user should verify that the module is operating in the desired mode of operation (FIPS-Approved mode or non-Approved mode) by observing the command prompt when logged into the device. If the string ":fips" is present then the device is operating in a FIPS-Approved mode. Otherwise it is operating in a non-Approved mode.

All FIPS users, including the Crypto Officer, must observe security guidelines at all times.

All FIPS users must:

- Keep all passwords confidential.
- Store devices and documentation in a secure area.
- Deploy devices in secure areas.
- Check audit files periodically.
- Conform to all other FIPS 140-2 security rules.
- Follow below guidelines:
    - Users are trusted.
    - Users abide by all security guidelines.
    - Users do not deliberately compromise security.
    - Users behave responsibly at all times.

# 7 References and Definitions

The following standards are referred to in this Security Policy.

**Table 16 - References**

| Abbreviation | Full Specification Name |
|---|---|
| [FIPS140-2] | *Security Requirements for Cryptographic Modules*, May 25, 2001 |
| [SP800-131A] | *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011 |
| [IG] | *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program* |

**Table 17 - Acronyms and Definitions**

| Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| DH | Diffie-Hellman |
| DSA | Digital Signature Algorithm |
| ECDH | Elliptic Curve Diffie-Hellman |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EMC | Electromagnetic Compatibility |
| FIPS | Federal Information Processing Standard |
| HMAC | Keyed-Hash Message Authentication Code |
| MD5 | Message Digest 5 |
| RE | Routing Engine |
| RSA | Public-key encryption technology developed by RSA Data Security, Inc. |
| SCB | Switch Control Board |
| SHA | Secure Hash Algorithms |
| SSH | Secure Shell |
| Triple-DES | Triple - Data Encryption Standard |

**Table 18 - Datasheets**

| Model | Title | URL |
|---|---|---|
| MX240<br>MX480<br>MX960 | MX240, MX480, MX960 3D Universal Edge Routers | https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000597-en.pdf |
| MX2010<br>MX2020 | MX2000 3D Universal Edge Routers | https://www.juniper.net/assets/us/en/local/pdf/datasheets/1000417-en.pdf |
| EX9200 | EX9200 Ethernet Switch | https://www.juniper.net/us/en/local/pdf/datasheets/1000432-en.pdf |