JUNIPER OPENSSL CRYPTOGRAPHIC MODULE
VERSION 1.0

FIPS 140-2 NON-PROPRIETARY SECURITY POLICY

VERSION 1.5

Last update: 2021-06-22

Prepared by:

atsec information security corporation

9130 Jollyville Road, Suite 260

Austin, TX 78759

www.atsec.com

1 Cryptographic Module Specification

This document is the non-proprietary FIPS 140-2 Security Policy for version 1.0 of the Juniper OpenSSL Cryptographic Module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-2 (Federal Information Processing Standards Publication 140-2) for a Security Level 1 software module.

The following sections describe the cryptographic module and how it conforms to the FIPS 140-2 specification in each of the required areas.

1.1 Module Overview

The Juniper OpenSSL Cryptographic Module (hereafter referred to as "the module") is a set of software libraries implementing the Transport Layer Security (TLS) protocol v1.0, v1.1 and v1.2 and Datagram Transport Layer Security (DTLS) protocol v1.0 and v1.2, as well as general purpose cryptographic algorithms. The module provides cryptographic services to applications running in the user space of the underlying Linux operating system through a C language Application Program Interface (API). The module utilizes processor instructions to optimize and increase performance. The module can act as a TLS server or client, and interacts with other entities via TLS/DTLS network protocols.

For the purpose of the FIPS 140-2 validation, the module is a software-only, multi-chip standalone cryptographic module validated at overall security level 1. The table below shows the security level claimed for each of the eleven sections that comprise the FIPS 140-2 standard.

Table 1 - Security Levels

FIPS 140-2 Section		Security Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	1
Overall Level		1

The cryptographic logical boundary consists of all shared libraries and the integrity check files used for Integrity Tests. The following table enumerates the files that comprise the module.

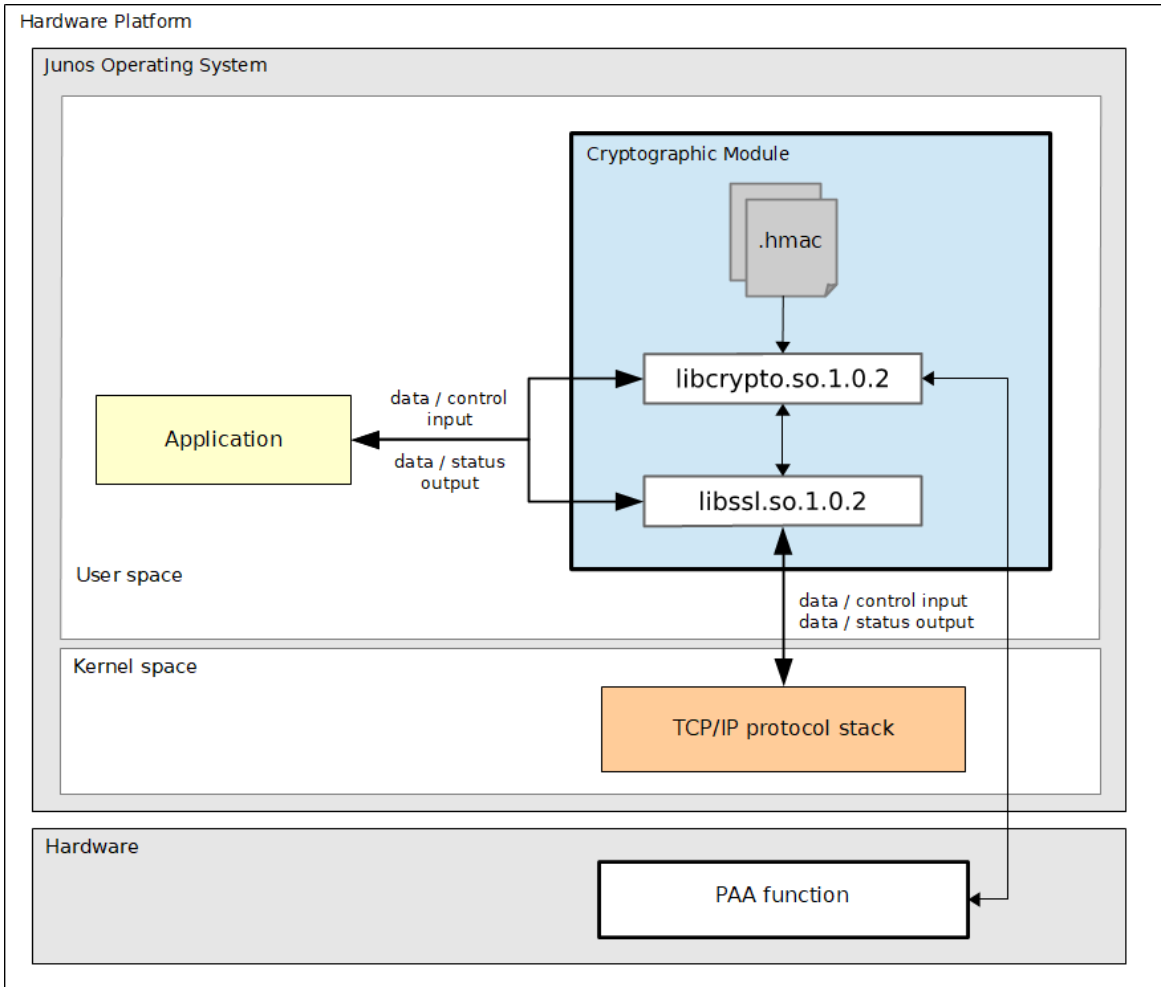
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Table 2 - Module Components

Component	Description
/usr/lib64/libssl.so.1.0.2	Shared library for TLS/DTLS network protocols.
/usr/lib64/libcrypto.so.1.0.2	Shared library for cryptographic implementations.
/usr/lib64/.libssl.so.1.0.2.hmac	Integrity check signature for libssl shared library.
/usr/lib64/.libcrypto.so.1.0.2.hmac	Integrity check signature for libcrypto shared library.

The software block diagram below shows the module, its interfaces with the operational environment and the delimitation of its logical boundary, comprised of all the components within the **BLUE** box.

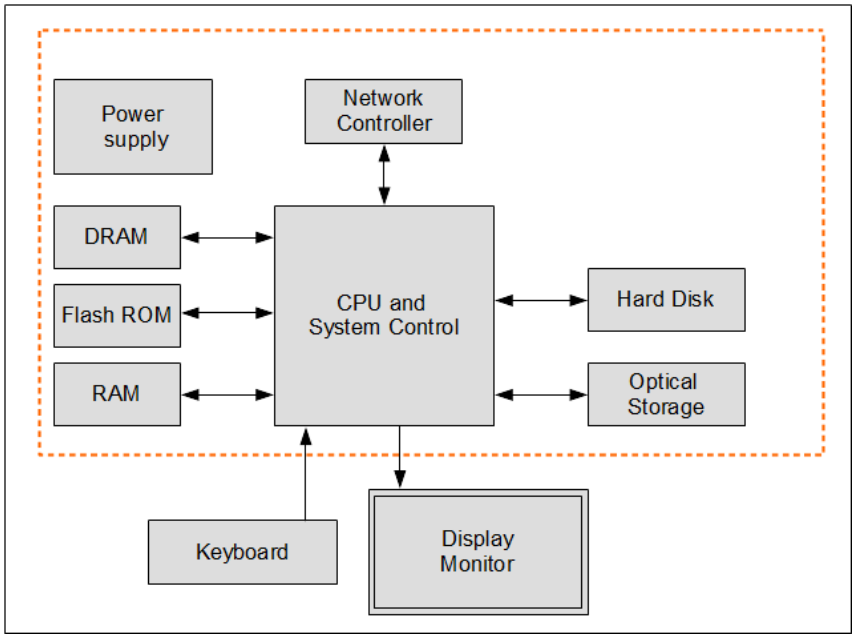
Figure 1 - Software Block Diagram



The module is aimed to run on a general purpose computer (GPC); the physical boundary of the module is the tested platform. Figure 2 shows the major components of a GPC.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Figure 2 - Hardware Block Diagram



The module has been tested on the test platforms shown below.

Table 3 - Tested Platforms

Test Platform	Processor	Test Configuration
Juniper Networks® Packet Transport Router Model PTX10003-80C	Intel® Xeon® E5-2628L v4	Junos OS Evolved version 19.4R2 with and without AES-NI (PAA)

Note: Per FIPS 140-2 IG G.5, the Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module or the security strengths of the generated keys when this module is ported and executed in an operational environment not listed on the validation certificate.

1.2 Modes of Operation

The module supports two modes of operation:

- **FIPS mode** (the Approved mode of operation): only approved or allowed security functions with sufficient security strength can be used.
- **non-FIPS mode** (the non-Approved mode of operation): only non-approved security functions can be used.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

The module enters FIPS mode after power-up tests succeed. Once the module is operational, the mode of operation is implicitly assumed depending on the security function invoked and the security strength of the cryptographic keys.

Critical security parameters used or stored in FIPS mode are not used in non-FIPS mode, and vice versa.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

o +1 408 745 2000
f +1 408 745 2100

2 Cryptographic Module Ports and Interfaces

As a software-only module, the module does not have physical ports. For the purpose of the FIPS 140-2 validation, the physical ports are interpreted to be the physical ports of the hardware platform on which it runs.

The logical interfaces are the API through which applications request services, and the TLS protocol internal state and messages sent and received from the TCP/IP protocol. The following table summarizes the four logical interfaces.

Table 4 - Ports and Interfaces

FIPS Interface	Logical Interface
Data Input	API input parameters, kernel I/O – network or files on file system, TLS protocol input messages.
Data Output	API output parameters, kernel I/O – network or files on file system, TLS protocol output messages.
Control Input	API function calls, API input parameters for control.
Status Output	API return codes, error messages.
Power Input	N/A

Note: The module is an implementation of the TLS protocol as defined in the RFC standards. The TLS protocol provides confidentiality and data integrity between communicating applications. When an application calls into the module's API, the data passed will be securely passed to the peer.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

3 Roles, Services and Authentication

3.1 Roles

The module supports the following roles:

- **User role**: performs all services (in both FIPS mode and non-FIPS mode) with the exception of module installation and configuration, and certificate management.
- **Crypto Officer role**: performs module installation and configuration, and certificate management.

The User and Crypto Officer roles are implicitly assumed by the entity accessing the module services.

3.2 Services

The module provides services to users that assume one of the available roles. All services are shown in Table 5 and Table 6, and described in detail in the user documentation (i.e., man pages).

The table below shows the services available in FIPS mode. For each service, the associated cryptographic algorithms, the roles to perform the service, and the cryptographic keys or Critical Security Parameters (CSPs) and their access rights are listed. The following convention is used to specify access rights to a CSP:

- **Create**: the calling application can create a new CSP.
- **Read**: the calling application can read the CSP.
- **Update**: the calling application can write a new value to the CSP.
- **Zeroize**: the calling application can zeroize the CSP.
- **n/a**: the calling application does not access any CSP or key during its operation.

If the services involve the use of the cryptographic algorithms, the corresponding Cryptographic Algorithm Validation Program (CAVP) certificate numbers of the cryptographic algorithms can be found in Table 7 of this security policy. Notice that the algorithms mentioned in the Network Protocol Services correspond to the same implementation of the algorithms described in the Cryptographic Library Services.

Table 5 - Services in FIPS mode of operation

Service	Algorithms	Role	Access	Keys/CSP
Cryptographic Library Services				
Symmetric Encryption and Decryption	AES	User	Read	AES key
RSA key generation	RSA, DRBG	User	Create	RSA public and private keys
RSA digital signature generation and verification	RSA	User	Read	RSA public and private keys
ECDSA key generation	ECDSA, DRBG	User	Create	ECDSA public and private keys
ECDSA public key validation	ECDSA	User	Read	ECDSA public key

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Service	Algorithms	Role	Access	Keys/CSP
ECDSA signature generation and verification	ECDSA	User	Read	ECDSA public and private keys
Random number generation	DRBG	User	Read, Update	Entropy input string, Internal state
Message digest	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	User	n/a	n/a
Message authentication code (MAC)	HMAC	User	Read	HMAC key
	CMAC with AES	User	Read	AES key
Key wrapping	AES-KW	User	Read	AES key
Key Derivation	SSH KDF	User	Read	Shared secret
			Create	Derived keys
Other Services				
Show status	n/a	User	n/a	None
Zeroization	n/a	User	Zeroize	All CSPs
Self-Tests	AES, SHS, HMAC, RSA, ECDSA, DRBG	User	n/a	None
Module installation	n/a	Crypto Officer	n/a	None
Module configuration	n/a	Crypto Officer	n/a	None

The table below lists the services only available in non-FIPS mode of operation.

Table 6 - Services in non-FIPS mode of operation

Service	Algorithms / Key sizes	Role	Access	Keys
Symmetric encryption and decryption	Algorithms listed in Table 9	User	Read	Symmetric key
Symmetric decryption	2-key Triple-DES listed in Table 9	User	Read	2-key Triple-DES key
Authenticated Encryption cipher for encryption and decryption	AES and SHA from multi-buffer or stitch implementation listed in Table 9	User	Read	AES key, HMAC key
Message digest	Algorithms listed in Table 9	User	n/a	None
Message authentication	HMAC and CMAC	User	Read	HMAC key, 2-key Triple-DES key

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

code (MAC)	restrictions listed in Table 9			
Domain Parameter Generation	DSA	User	Read	N/A
Key Pair generation	RSA, ECDSA restrictions listed in Table 9	User	Create	Asymmetric public and private keys
	DSA with any key sizes			
Public Key Validation	ECDSA restrictions listed in Table 9	User	Read	Asymmetric public and private keys
Digital signature generation	RSA, ECDSA restrictions listed in Table 9	User	Read	Asymmetric public and private keys
	DSA with any key sizes	User	Read	
Digital signature Verification	RSA, ECDSA and message digest restrictions listed in Table 9	User	Read	Asymmetric public and private keys
	DSA with any key sizes	User	Read	
Key agreement	SRP, J-PAKE	User	Create, Read	Asymmetric public and private keys
Key agreement	Diffie-Hellman, EC Diffie-Hellman	User	Create, Read	Diffie-Hellman, EC Diffie-Hellman public and private keys
			Create	Shared secret
Shared secret computation	Diffie-Hellman, EC Diffie-Hellman	User	Create, Read	Diffie-Hellman, EC Diffie-Hellman public and private keys
			Create	Shared secret
Key encapsulation	RSA	User	Read	RSA public and private keys
Key Derivation	TLS KDF	User	Read	Shared secret
			Create	Derived keys
Transport Layer Security (TLS) network protocol v1.0, v1.1 and v1.2	RSA, DSA, ECDSA	User	Read	RSA, DSA or ECDSA public and/or private keys
	Diffie-Hellman, EC Diffie-Hellman, AES, Triple-DES, HMAC, TLS KDF	User	Create, Read	TLS pre_master_secret, TLS master_secret, Diffie Hellman or EC Diffie Hellman public and private keys, AES or Triple-DES key, HMAC key
TLS extensions	n/a	User	Read	RSA, DSA or ECDSA public and/or private keys
Certificate management	n/a	Crypto Officer	Read	RSA, DSA or ECDSA public and/or private keys

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

3.3 Algorithms

The Juniper OpenSSL Cryptographic Module is compiled to use the support from the processor and assembly code for AES, SHA and GHASH operations to enhance the performance of the module. Different implementations can be invoked by using a processor capability mask in the operational environment. Please note that only one AES, SHA and/or GHASH implementation can be executed in runtime.

The module supports the use of AES-NI, SSSE3 and strict assembler for AES implementations, the use of AVX2, SSSE3 and strict assembler for SHA implementations, and the use of CLMUL instruction set and strict assembler for GHASH that is used for GCM mode. The module uses the most efficient implementation based on the processor's capability. In the case of the tested environment, only the implementations for AES, SHA-1 and SHA-2 using either AES-NI (PAA) and strict assembler (non PAA) have been CAVP tested. Therefore, these are the only approved functions.

The following table shows the CAVP certificates and their associated information of the cryptographic implementation in FIPS mode.

Table 7 - Cryptographic Algorithms

Algorithm	Mode / Method	Key Lengths, Curves or Moduli (in bits)	Use	Standard	CAVP Certs
AES	ECB, CBC, OFB, CFB1, CFB8, CFB128, CTR	128, 192, 256	Data Encryption and Decryption	[FIPS197], [SP800-38A]	A650 A2017
	CMAC	128, 192, 256	MAC Generation and Verification	[SP800-38B]	A650 A2017
	CCM	128, 192, 256	Data Encryption and Decryption	[SP800-38C]	A650 A2017
	XTS	128, 256	Data Encryption and Decryption for Data Storage	[SP800-38E]	A650 A2017
	KW	128, 192, 256	Key Wrapping and Unwrapping	[SP800-38F]	A650 A2017
DRBG	Hash_DRBG: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 without PR	n/a	Deterministic Random Bit Generation	[SP800-90A]	A650
	HMAC_DRBG: SHA-1, SHA-224, SHA-256,				A650

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Algorithm	Mode / Method	Key Lengths, Curves or Moduli (in bits)	Use	Standard	CAVP Certs
	SHA-384, SHA-512 without PR				
	CTR_DRBG: AES-128, AES-192, AES-256 with/without DF, without PR				A650
ECDSA		P-256, P-384, P-521	Key Pair Generation	[FIPS186-4]	A650
		P-192, P-224, P-256, P-384, P-521	Public Key Verification	[FIPS186-4]	A650
	SHA-224, SHA-256, SHA-384, SHA-512	P-224, P-256, P-384, P-521	Digital Signature Generation	[FIPS186-4]	A650
	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	P-192, P-224, P-256, P-384, P-521	Digital Signature Verification	[FIPS186-4]	A650
HMAC	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	112 or greater	Message Authentication Code	[FIPS198-1]	A650
SSH v2 KDF	SHA-1, SHA-256, SHA-384, SHA-512	n/a	Key Derivation in the SSHv2 protocol	[SP800-135]	CVL. A650
RSA	B.3.3	2048, 3072, 4096	Key Pair Generation	[FIPS186-4]	A650
	X9.31 with SHA-256, SHA-384, SHA-512	2048, 3072, 4096	Digital Signature Generation	[FIPS186-4]	A650
	X9.31 with SHA-1, SHA-256, SHA-384, SHA-512	1024, 2048, 3072, 4096	Digital Signature Verification	[FIPS186-4]	A650
	PKCS#1v1.5, PSS with SHA-224,	2048, 3072, 4096	Digital Signature Generation	[FIPS186-4]	A650

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Algorithm	Mode / Method	Key Lengths, Curves or Moduli (in bits)	Use	Standard	CAVP Certs
	SHA-256, SHA-384, SHA-512				
	PKCS#1v1.5, PSS with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1024, 2048, 3072, 4096	Digital Signature Verification	[FIPS186-4]	A650
SHS	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	n/a	Message Digest	[FIPS180-4]	A650 A2017
KTS	AES KW	128, 192, 256	Key Wrapping and unwrapping	SP800-38F	A650 A2017

3.3.1 Allowed Algorithms

The following table describes the non-Approved but allowed algorithms in FIPS mode:

Table 8 – FIPS-Allowed Cryptographic Algorithms

Algorithm	Caveat	Use
NDRNG	n/a	The module obtains the entropy data from NDRNG to seed the DRBG.

3.3.2 Non-Approved Algorithms

The table below shows the non-Approved cryptographic algorithms implemented in the module that are only available in non-FIPS mode.

Table 9 - Non-Approved Cryptographic Algorithms

Algorithm	Use
Blowfish, Camellia, CAST, DES, IDEA, RC2, RC4, RC5 and SEED	Data Encryption and Decryption
AES in XTS mode with 192-bit key	Data Encryption and Decryption
Triple-DES	Data Encryption and Decryption
MD2, MD4, MD5, MDC-2, RMD160, Whirpool	Message Digest

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Algorithm	Use
J-PAKE	Password Authenticated Key Exchange
RSA with key size smaller than 2048 bits or greater than 4096 bits	Key Pair Generation, Digital Signature Generation
RSA with key size smaller than 1024 bits or greater than 4096 bits	Digital Signature Verification
RSA Key Encapsulation with Encryption and Decryption Primitives	Key Establishment;
DSA with any key sizes	Key Pair Generation, Domain Parameter Generation, Digital Signature Generation, Digital Signature Verification
ECDSA with curve P-192	Key Pair Generation, Public Key Validation, Digital Signature Generation
ECDSA with curve P-224	Key Pair Generation
ECDSA with K and B curves, and non-NIST curves	Key Pair Generation, Public Key Validation, Digital Signature Generation and Verification
Diffie-Hellman with any key sizes	Key Agreement, Shared Secret Computation
EC Diffie-Hellman with any curve.	Key Agreement, Shared Secret Computation
SHA-1	Digital Signature Generation
AES-GMAC	Message Authentication Code
HMAC with less than 112 bits key	Message Authentication Code
CMAC with 2-key Triple-DES	Message Authentication Code
SRP	Key Agreement
Multiblock ciphers using AES in CBC mode with 128 and 256 bit keys and HMAC SHA-1 and SHA-256	Authenticated Data Encryption and Decryption
SSLeay Deterministic Random Number Generator (PRNG)	Random Number Generation
AES, SHA-1 and SHA-2 algorithm implementations not using AES-NI or strict assembler.	All services where these algorithms are involved.
TLS v1.0, v1.1 and v1.2 KDF	Key Derivation in the TLS protocol
MD5	Pseudo-random function (PRF) in TLS v1.0 and v1.1
AES CCM	Key Wrapping and Unwrapping for the TLS protocol.
AES GCM	
AES CBC and HMAC	
Triple-DES CBC and HMAC	

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Algorithm	Use
AES GCM	Data Encryption and Decryption
RSA and ECDSA	Signature operation in the context of TLS protocol

3.4 Operator Authentication

The module does not implement user authentication. The role of the user is implicitly assumed based on the service requested.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

4 Physical Security

The module is comprised of software only and therefore this security policy does not make any claims on physical security.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

5 Operational Environment

5.1 Applicability

The module operates in a modifiable operational environment per FIPS 140-2 level 1 specifications. The module runs on a commercially available general-purpose operating system executing on the hardware specified in Table 3.

5.2 Policy

The operating system is restricted to a single operator; concurrent operators are explicitly excluded.

The application that requests cryptographic services is the single user of the module.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

6 Cryptographic Key Management

The following table summarizes the Critical Security Parameters (CSPs) that are used by the cryptographic services implemented in the module:

Table 10 - Lifecycle of Critical Security Parameters (CSPs)

Name	Generation	Entry and Output	Zeroization
AES keys	Key material is entered via API parameter	The key is passed into the module via API input parameters in plaintext.	EVP_CIPHER_CTX_cleanup()
HMAC keys			HMAC_CTX_cleanup()
RSA public and private keys	Public and private keys are generated using the FIPS 186-4 key Generation method; random values are obtained using the SP800 90A DRBG.	The key is passed into the module via API input parameters in plaintext. The key is passed out of the module via API output parameters in plaintext.	RSA_free()
ECDSA public and private keys			EC_KEY_free()
Derived key	Generated during SSH KDF.	Keys are passed out of the module via API output parameters in plaintext.	EVP_PKEY_free()
Entropy input string	Obtained from the NDRNG.	None	FIPS_drbg_free()
DRBG internal state (V, C, Key)	During DRBG initialization.	None	FIPS_drbg_free()

The following sections describe how CSPs, in particular cryptographic keys, are managed during its life cycle.

6.1 Random Number Generation

The module employs a Deterministic Random Bit Generator (DRBG) based on [SP800-90A] for the creation of key components of asymmetric keys. In addition, the module provides a Random Number Generation service to calling applications.

The DRBG supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG mechanisms. The DRBG is initialized during module initialization; the module loads by default the DRBG using the CTR_DRBG mechanism with AES-256, with derivation function without prediction resistance. A different DRBG mechanism can be chosen through an API function call.

The module uses a Non-Deterministic Random Number Generator (NDRNG) as the entropy source for seeding the DRBG. The NDRNG is provided by the operational environment (i.e., the Linux RNG via /dev/urandom), which is within the module's physical boundary but outside of the module's logical boundary. The NDRNG provides at least 128 bits of entropy to the DRBG during initialization (seed) and reseeding (reseed).

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

The Linux kernel performs conditional self-tests on the output of NDRNG to ensure that consecutive random numbers do not repeat. The module performs the DRBG health tests as defined in section 11.3 of [SP800-90A].

CAVEAT: The module generates cryptographic keys whose strengths are modified by available entropy.

6.2 Key Generation

The Module provides an SP800-90A-compliant Deterministic Random Bit Generator (DRBG) for creation of key components of asymmetric keys, and random number generation.

The Key Generation methods implemented in the module for Approved services in FIPS mode is compliant with [SP800-133].

For generating RSA and ECDSA keys the module implements asymmetric key generation services compliant with [FIPS186-4]. A seed (i.e. the random value) used in asymmetric key generation is directly obtained from the [SP800-90A] DRBG.

6.3 Key Agreement / Key Transport / Key Derivation

The module does not provide any key agreement in FIPS approved mode of operation.

The module provides the following key transport mechanisms:

- Key wrapping using AES-KW.

According to Table 2: Comparable strengths in [SP 800-57], the key sizes of AES provides the following security strength in FIPS mode of operation:

- AES key wrapping using AES in KW provides between 128 and 256 bits of encryption strength.

The module supports the following key derivation methods according to [SP800-135]:

- KDF for the SSHv2 protocol.

6.4 Key Entry / Output

The module does not support manual key entry or intermediate key generation key output. The keys are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form. This is allowed by [FIPS140-2_IG] IG 7.7, according to the “CM Software to/from App Software via GPC INT Path” entry on the Key Establishment Table.

6.5 Key / CSP Storage

Symmetric keys, HMAC keys, public and private keys are provided to the module by the calling application via API input parameters and are destroyed by the module when invoking the appropriate API function calls.

The module does not perform persistent storage of keys. The keys and CSPs are stored as plaintext in the RAM. The only exception is the HMAC key used for the Integrity Test, which is stored in the module and relies on the operating system for protection.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

6.6 Key / CSP Zeroization

The memory occupied by keys is allocated by regular memory allocation operating system calls. The application is responsible for calling the appropriate zeroization functions provided in the module's API listed in Table 10. The zeroization functions overwrite the memory occupied by keys with “zeros” and deallocate the memory with the regular memory deallocation operating system call.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

7 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The test platforms listed in Table 3 have been tested and found to conform to the EMI/EMC requirements specified by 47 Code of Federal Regulations, FCC PART 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., Business use). These devices are designed to provide reasonable protection against harmful interference when the devices are operated in a commercial environment. They shall be installed and used in accordance with the instruction manual.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

8 Self-Tests

FIPS 140-2 requires that the module perform power-up tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, some functions require continuous testing of the cryptographic functionality, such as the asymmetric key generation. If any self-test fails, the module returns an error code and enters the error state. No data output or cryptographic operations are allowed in error state.

See section 9.3 for descriptions of possible self-test errors and recovery procedures.

8.1 Power-Up Tests

The module performs power-up tests when the module is loaded into memory without operator intervention. Power-up tests ensure that the module is not corrupted and that the cryptographic algorithms work as expected.

While the module is executing the power-up tests, services are not available, and input and output are inhibited. The module is not available for use by the calling application until the power-up tests are completed successfully.

If any power-up test fails, the module returns the error code listed in Table 13 and displays the specific error message associated with the returned error code, and then enters error state. The subsequent calls to the module will also fail - thus no further cryptographic operations are possible. If the power-up tests complete successfully, the module will return 1 in the return code and will accept cryptographic operation service requests.

8.1.1 Integrity Tests

The integrity of the module is verified by comparing an HMAC-SHA-256 value calculated at run time with the HMAC value stored in the .hmac file that was computed at build time for each software component of the module. If the HMAC values do not match, the test fails and the module enters the error state.

8.1.2 Cryptographic Algorithm Tests

The module performs self-tests on all FIPS-Approved cryptographic algorithms supported in the Approved mode of operation, using the Known Answer Tests (KAT) and Pair-wise Consistency Tests (PCT) shown in the following table:

Table 11 - Self-tests[3]

Algorithm	Power-Up Tests
AES	• KAT AES ECB mode with 128-bit key, encryption and decryption (separately tested) • KAT AES CCM mode with 192-bit key, encryption and decryption (separately tested) • KAT AES XTS mode with 128 and 256 bit key, encryption and decryption (separately tested)

[3] Additionally, the module implements KATs for Diffie-Hellman/EC Diffie-Hellman primitive "Z" computation, Triple-DES encryption, RSA encrypt/decrypt and AES GCM as well as DSA PCT as part of power-on self-tests. However, these self-tests are not listed here as these algorithms are non-Approved.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

Algorithm	Power-Up Tests
CMAC	• KAT AES CMAC with 128, 192 and 256 bit keys, MAC generation
SHS	• KAT SHA-1, SHA-256 and SHA-512
HMAC	• KAT HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512
ECDSA	• PCT ECDSA with P-256 and SHA-256
RSA	• KAT RSA PKCS#1 v1.5 scheme with 2048-bit key, using SHA-1, SHA-224, SHA 256, SHA-384 and SHA-512, signature generation and verification (separately tested) • KAT RSA PSS scheme with 2048-bit key, using SHA-1, SHA-224, SHA 256, SHA-384 and SHA-512, signature verification and verification (separately tested)
DRBG	• KAT CTR_DRBG with AES with 256-bit key, without PR, with DF • KAT CTR_DRBG with AES with 256-bit key, without PR, without DF • KAT Hash_DRBG with SHA-256, without PR • KAT HMAC_DRBG with SHA-256, without PR

For the KAT, the module calculates the result and compares it with the known value. If the answer does not match the known answer, the KAT is failed and the module enters the Error state.

For the PCT, if the signature generation or verification fails, the module enters the Error state. As described in section 3.3, only one AES or SHA implementation is available at run-time.

8.2 On-Demand Self-Tests

On-Demand self-tests can be invoked by powering-off and reloading the module which cause the module to run the power-up tests again. During the execution of the on-demand self-tests, services are not available and no data output or input is possible.

8.3 Conditional Tests

The module performs conditional tests on the cryptographic algorithms, using the Pair-wise Consistency Tests (PCT) and Continuous Random Number Generator Test (CRNGT), shown in the following table:

Table 12 - Conditional Tests[4]

Algorithm	Conditional Test
ECDSA key generation	• PCT using SHA-256, signature generation and verification.
RSA key generation	• PCT using SHA-256, signature generation and verification.
DRBG	• CRNGT is not required per IG 9.8

[4] Additionally, the module implements DSA key generation PCT and RSA PCT with encrypt/decrypt but these are not listed here as these algorithm are non-Approved.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

NDRNG	• CRNGT (not performed by the module, but by the underlying Linux Operating System)

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

o +1 408 745 2000
f +1 408 745 2100

9 Guidance

9.1 Crypto Officer Guidance

The binaries of the module are contained in the base Junos Evolved installation image. The Crypto Officer shall follow this Security Policy to configure the operational environment and install the module to be operated as a FIPS 140-2 validated module.

9.1.1 Operating Environment Configurations

To configure the operating environment to support FIPS, the following shall be performed with the root privilege:

(1) Enter CLI configuration mode.

(2) Commit changes:
```
commit
```

(3) Configure FIPS level to 1:
```
set system fips level 1
```

(4) Commit changes:
```
commit
```

(5) Exit configuration mode to enter operational mode:
```
exit
```

(6) Reboot the system with the new settings (answer `yes` to prompt):
```
request system reboot
```

Now, the operating environment is configured to support FIPS operation. The Crypto Officer should check the existence of the file, `/proc/sys/crypto/fips_enabled`, and that it contains "1". If the file does not exist or does not contain "1", the operating environment is not configured to support FIPS and the module will not operate as a FIPS validated module properly.

9.2 User Guidance

In order to run in FIPS mode, the module must be operated using the FIPS Approved services, with their corresponding FIPS Approved and FIPS allowed cryptographic algorithms provided in this Security Policy (see section 3.2). In addition, key sizes must comply with [SP800-131A].

9.2.1 AES XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in [SP800-38E]. The length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks that is 16MB of data. To meet the requirement in [FIPS140-2_IG] A.9, the module implements a check to ensure that the two AES keys used in XTS-AES algorithm are not identical.

Note: AES-XTS shall be used with 128 and 256-bit keys only. AES-XTS with 192-bit keys is not an Approved service.

9.2.2 Random Number Generator

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

The **RAND_cleanup()** API function must not be used. This call will clean up the internal DRBG state. This call also replaces the DRBG instance with the non-FIPS Approved SSLeay Deterministic Random Number Generator when using the **RAND_*** API calls.

9.2.3 API Functions

Passing "0" to the **FIPS_mode_set()** API function is prohibited.

Executing the **CRYPTO_set_mem_functions()** API function is prohibited as it performs like a null operation in the module.

9.2.4

9.2.4 Environment Variables

OPENSSL_ENFORCE_MODULUS_BITS

As described in [SP800-131A], less than 2048 bits of RSA key sizes are disallowed by NIST. Setting the environment variable **OPENSSL_ENFORCE_MODULUS_BITS** can restrict the module to only generate the acceptable key sizes of RSA. If the environment variable is set, the module can generate RSA keys with 2048 bits or more (actually only keys with 2048, 3072 and 4096 bits are allowed in FIPS mode of operation).

OPENSSL_ia32cap

This capability mask environment variable can be used to remove one or more processor capabilities, forcing the module to use different (and less efficient) algorithm implementations for AES and SHA. Altering the processor capabilities is not recommended.

Should the environment variable be used to mask off the PAA implementations, the only value allowed in the approved mode of operation is the following:

```
OPENSSL_ia32cap="~0x1200020200000000:~0x0020000020"
```

9.3 Handling FIPS Related Errors

When the module fails any self-test, the module will return an error code to indicate the error and enters error state that any further cryptographic operation is inhibited. Errors occurred during self-tests and conditional tests forces the module to transition to an error state. Here is the list of error codes when the module fails any self-test, in error state or not supported in FIPS mode:

Table 13 - Error Events, Error Codes and Error Messages

Error Events	Error Codes/Messages
When the Integrity Test fails at the power-up	FIPS_R_FINGERPRINT_DOES_NOT_MATCH (111) "fingerprint does not match"
When the AES, SHA-1, SHA-512 KAT fails at the power-up	FIPS_R_SELFTEST_FAILED (134) "selftest failed"
When the KAT for RSA fails, or the PCT for ECDSA or DSA[5] fails at the power-up	FIPS_R_TEST_FAILURE (137)

[5] The module implements DSA key generation PCT although this algorithm is non-Approved.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

	"test failure"
When the KAT of DRBG fails at the power-up	FIPS_R_NOPR_TEST1_FAILURE (145) "nopr test1 failure"
When the new generated RSA or ECDSA key pair fails the PCT	FIPS_R_PAIRWISE_TEST_FAILED (127) "pairwise test failed"
When the module is in error state and any cryptographic operation is called	FIPS_R_FIPS_SELFTEST_FAILED (115) "fips selftest failed"
	FIPS_R_SELFTEST_FAILED (134) "selftest failed"
When the AES key and tweak keys for XTS-AES are the same	FIPS_R_AES_XTS_WEAK_KEY (201) "identical keys are weak"

These errors are reported through the regular **ERR** interface of the modules and can be queried by functions such as **ERR_get_error()**. See the OpenSSL man pages for the function description.

When the module is in the error state and the application calls a crypto function of the module that cannot return an error in normal circumstances (void return functions), the error message: "**OpenSSL internal error, assertion failed: FATAL FIPS SELFTEST FAILURE**" is printed to **stderr** and the application is terminated with the **abort()** call. The only way to recover from this error is to restart the application. If the failure persists, the module must be reinstalled.

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

10 Mitigation of Other Attacks

10.1 Blinding Against RSA Timing Attacks

RSA is vulnerable to timing attacks. In a configuration where attackers can measure the time of RSA decryption or signature operations, blinding must be used to protect the RSA operation from that attack.

The module provides the API functions **RSA_blinding_on()** and **RSA_blinding_off()** to turn the blinding on and off for RSA. When the blinding is on, the module generates a random value to form a blinding factor in the RSA key before the RSA key is used in the RSA cryptographic operations.

Please note that the DRBG must be seeded prior to calling **RSA_blinding_on()** to prevent the RSA Timing Attack.

10.2 Weak Triple-DES Keys Detection

The module implements the **DES_set_key_checked()** for checking the weak Triple-DES key and the correctness of the parity bits when the Triple-DES key is going to be used in Triple-DES operations. The checking of the weak Triple-DES key is implemented in the API function **DES_is_weak_key()** and the checking of the parity bits is implemented in the API function **DES_check_key_parity()**. If the Triple-DES key does not pass the check, the module will return -1 to indicate the parity check error and -2 if the Triple-DES key matches to any value listed below:

```
static const DES_cblock weak_keys[NUM_WEAK_KEY] = {
    /* weak keys */
    {0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01},
    {0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE},
    {0x1F, 0x1F, 0x1F, 0x1F, 0x0E, 0x0E, 0x0E, 0x0E},
    {0xE0, 0xE0, 0xE0, 0xE0, 0xF1, 0xF1, 0xF1, 0xF1},
    /* semi-weak keys */
    {0x01, 0xFE, 0x01, 0xFE, 0x01, 0xFE, 0x01, 0xFE},
    {0xFE, 0x01, 0xFE, 0x01, 0xFE, 0x01, 0xFE, 0x01},
    {0x1F, 0xE0, 0x1F, 0xE0, 0x0E, 0xF1, 0x0E, 0xF1},
    {0xE0, 0x1F, 0xE0, 0x1F, 0xF1, 0x0E, 0xF1, 0x0E},
    {0x01, 0xE0, 0x01, 0xE0, 0x01, 0xF1, 0x01, 0xF1},
    {0xE0, 0x01, 0xE0, 0x01, 0xF1, 0x01, 0xF1, 0x01},
    {0x1F, 0xFE, 0x1F, 0xFE, 0x0E, 0xFE, 0x0E, 0xFE},
    {0xFE, 0x1F, 0xFE, 0x1F, 0xFE, 0x0E, 0xFE, 0x0E},
    {0x01, 0x1F, 0x01, 0x1F, 0x01, 0x0E, 0x01, 0x0E},
    {0x1F, 0x01, 0x1F, 0x01, 0x0E, 0x01, 0x0E, 0x01},
    {0xE0, 0xFE, 0xE0, 0xFE, 0xF1, 0xFE, 0xF1, 0xFE},
    {0xFE, 0xE0, 0xFE, 0xE0, 0xFE, 0xF1, 0xFE, 0xF1}
};
```

11
11

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

11 Appendix B - Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Program Interface
APT	Advanced Package Tool
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CLMUL	Carry-less Multiplication
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPACF	CP Assist for Cryptographic Function
CRNGT	Continuous Random Number Generator Test
CSP	Critical Security Parameter
CTR	Counter Mode
DES	Data Encryption Standard
DF	Derivation Function
DSA	Digital Signature Algorithm
DTLS	Datagram Transport Layer Security
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
EMI/EMC	Electromagnetic Interference/Electromagnetic Compatibility
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
GPC	General Purpose Computer
HMAC	Hash Message Authentication Code
IG	Implementation Guidance
KAS	Key Agreement Schema
KAT	Known Answer Test
KDF	Key Derivation Function
KW	Key Wrap
LPAR	Logical Partitions

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NDRNG	Non-Deterministic Random Number Generator
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PCT	Pair-wise Consistency Test
PR	Prediction Resistance
PRNG	Pseudo-Random Number Generator
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSSE3	Supplemental Streaming SIMD Extensions 3
TLS	Transport Layer Security
XTS	XEX-based Tweaked-codebook mode with ciphertext Stealing

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

12 Appendix C - References

FIPS140-2 **FIPS PUB 140-2 - Security Requirements For Cryptographic Modules**
May 2001
http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

FIPS140-2_IG **Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program**
August 12, 2020
http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

FIPS180-4 **Secure Hash Standard (SHS)**
March 2012
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

LMAN **Linux Man Pages**
http://man7.org/linux/man-pages/

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1**
February 2003
http://www.ietf.org/rfc/rfc3447.txt

RFC2246 **The TLS Protocol Version 1.0**
January 1999
https://www.ietf.org/rfc/rfc2246.txt

RFC3268 **Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)**
June 2002
https://www.ietf.org/rfc/rfc3268.txt

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

o +1 408 745 2000
f +1 408 745 2100

RFC4279	**Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)** December 2005 https://www.ietf.org/rfc/rfc4279.txt
RFC4346	**The Transport Layer Security (TLS) Protocol Version 1.1** April 2006 https://www.ietf.org/rfc/rfc4346.txt
RFC4492	**Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)** May 2006 https://www.ietf.org/rfc/rfc4492.txt
RFC5116	**An Interface and Algorithms for Authenticated Encryption** January 2008 https://www.ietf.org/rfc/rfc5116.txt
RFC5246	**The Transport Layer Security (TLS) Protocol Version 1.2** August 2008 https://tools.ietf.org/html/rfc5246.txt
RFC5288	**AES Galois Counter Mode (GCM) Cipher Suites for TLS** August 2008 https://tools.ietf.org/html/rfc5288.txt
RFC5487	**Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode** March 2009 https://tools.ietf.org/html/rfc5487.txt
RFC6655	**AES-CCM Cipher Suites for Transport Layer Security (TLS)** July 2012 https://tools.ietf.org/html/rfc6655.txt
RFC7251	**AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS** June 2014 https://tools.ietf.org/html/rfc7251.txt
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net

o +1 408 745 2000
f +1 408 745 2100

SP800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
SP800-38C	**NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-52	**NIST Special Publication 800-52 Revision 1 - Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations** April 2014 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf
SP800-56B	**NIST Special Publication 800-56B Revision 1 - Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography** September 2014 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br1.pdf
SP800-57	**NIST Special Publication 800-57 Part 1 Revision 4 - Recommendation for Key Management Part 1: General** January 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100

SP800-67	**NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** January 2012 http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf
SP800-90A	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-131A	**NIST Special Publication 800-131A – Revision 2 - Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf
SP800-135	**NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
www.juniper.net
o +1 408 745 2000
f +1 408 745 2100