Cisco Systems, Inc

CiscoSSL FIPS Provider

FIPS 140-3 Non-Proprietary Security Policy

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary Security Policy for the Cryptographic Module CiscoSSL FIPS Provider, firmware version 8.0. This Security Policy is provided in accordance with ISO/IEC 19790 Annex B, FIPS 140-3, and SP 800-140B. This Security Policy was prepared as part of the Level 1 FIPS 140-3 validation of the CiscoSSL FIPS provider, and the module meets the overall Level 1 requirements.

The following table lists the level of validation for each area in the FIPS PUB 140-3.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	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	1
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The CiscoSSL FIPS Provider is a firmware library that provides cryptographic services to a vast array of Cisco's networking and collaboration products. The cryptographic module provides the cipher operations and Key Derivation functions to support the following protocols: IKEv2/IPSec, sRTP, SSH, TLS, SNMPv3, ANS X9.42, and ANS X.9.63. Full implementations of these protocols are not supported by the module. No parts of the protocols, other than the KDF, have been tested by the CAVP or CMVP. The tested module version is 8.0. The overall security level is 1.

The object code in the object module file is incorporated into the runtime executable application at the time the binary executable is generated. The module is provided in an executable form

(as fips.so shared object). The module performs no communications other than with the consuming host application (the process that invokes the module services via the module's API), which can be considered as the host for the module.

Module Type: Firmware

Module Embodiment: Multi-Chip Standalone

Module Characteristics:

Cryptographic Boundary:

The cryptographic boundary of the module is the CiscoSSL FIPS Provider, a dynamically loadable library. The module is comprised of a single object module file called fips.so. The module performs no communication other than with the calling application via APIs that invoke the module.

Tested Operational Environment's Physical Perimeter (TOEPP):

The module's TOEPP is the physical perimeter of the tested platforms listed in Table "Tested Operational Environments - Software, Firmware, Hybrid" below. The components of the TOEPP include: Hardware components [Cisco UCS, Storage, RAM, Network Interface Cards].

The module's block diagram is shown in Figure 1 below. The dashed orange border in the figure denotes the cryptographic boundary of the module. The green border denotes the TOEPP of the module.

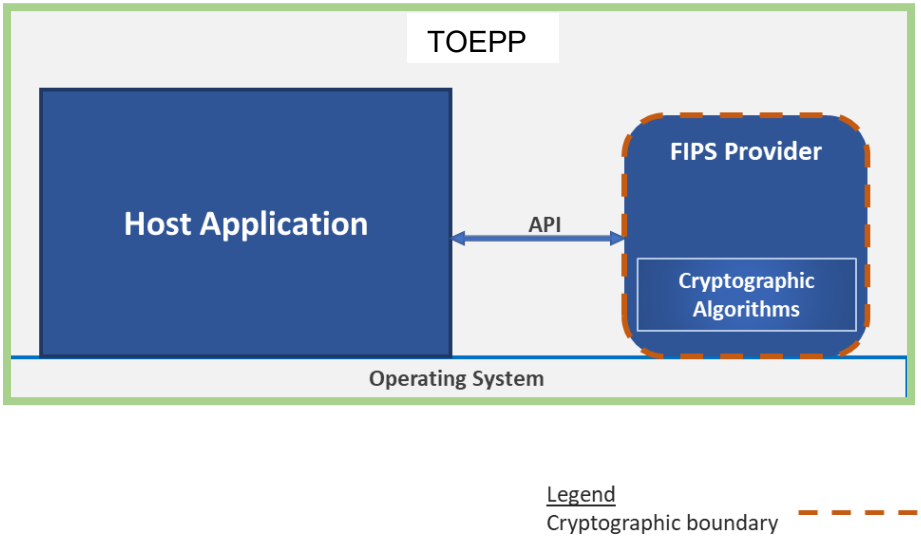

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.so	8.0		HMAC SHA2-256

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Cisco IOS-XE 17.14	Cisco Unified Computing System (UCS)	Intel Xeon Gold 6244	Yes	ESXi 7.0	8.0
Cisco IOS-XE 17.14	Cisco Unified Computing System (UCS)	Intel Xeon Gold 6244	No	ESXi 7.0	8.0

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

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

2.3 Excluded Components

There are no components excluded from the module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	Provides services approved by FIPS 140-3	Approved	Returns 1 when approved services are run successfully
Non-Approved Mode	Provides services not approved for use in FIPS 140-3	Non-Approved	Returns 3, ED25519, ED448, X25519, X448 when non-approved services are run successfully

Table 4: Modes List and Description

The module supports both approved and non-approved modes of operation. The module will only enter the approved mode if the module is reloaded and the call to SELF_TEST_post() succeeds, and only approved services are invoked. The module enters non-approved mode when a non-approved service is invoked.

Mode Change Instructions and Status:

When a non-approved service is invoked while in approved mode of operation, the module implicitly transitions to a non-approved mode. Similarly, when a call to an approved service is made while in non-approved mode of operation, the module transitions to approved mode of operation.

The mode can be identified by the indicator, as listed in the table “Modes List and Description” above.

2.5 Algorithms

Approved Algorithms:

PAA

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A3032	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-512 Increment 8	SP 800-38A
AES-CBC-CS2	A3032	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-512 Increment 8	SP 800-38A
AES-CBC-CS3	A3032	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 136-512 Increment 8	SP 800-38A
AES-CCM	A3032	Key Length - 128, 192, 256 Tag Length - 112, 128, 32, 48, 64, 80, 96 IV Length - IV Length: 56-104 Increment 8	SP 800-38C

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 0-256 Increment 8 AAD Length - AAD Length: 0-524288 Increment 8	
AES-CFB1	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3032	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-524288 Increment 8	SP 800-38B
AES-CTR	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3032	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96-1024 Increment 8 Payload Length - Payload Length: 8-65536 Increment 8 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
AES-GMAC	A3032	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
AES-KW	A3032	Direction - Decrypt, Encrypt Cipher - Cipher, Inverse Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F
AES-KWP	A3032	Direction - Decrypt, Encrypt Cipher - Cipher, Inverse Key Length - 128, 192, 256 Payload Length - Payload Length: 8-4096 Increment 8	SP 800-38F
AES-OFB	A3032	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-XTS Testing Revision 2.0	A3032	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
Counter DRBG	A3032	Prediction Resistance - Yes Supports Reseed - Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes Additional Input - Additional Input: 0-256 Increment 256, Additional Input: 256, Additional Input: 320, Additional Input: 384 Entropy Input - Entropy Input: 128-256 Increment 128, Entropy Input: 256, Entropy Input: 256-512 Increment 128, Entropy Input: 320, Entropy Input: 384 Nonce - Nonce: 0, Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 256, Personalization String Length: 256, Personalization String Length: 320, Personalization String Length: 384 Returned Bits - 256	SP 800-90A Rev. 1
DSA KeyGen (FIPS186-4)	A3032	L - 2048, 3072 N - 224, 256	FIPS 186-4
DSA PQGGen (FIPS186-4)	A3032	P/Q Generation Methods - Probable G Generation Methods - Canonical, Unverifiable L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA PQGVer (FIPS186-4)	A3032	P/Q Generation Methods - Probable G Generation Methods - Canonical, Unverifiable L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA SigGen (FIPS186-4)	A3032	L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA SigVer (FIPS186-4)	A3032	L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A3032	Curve - P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3032	Curve - P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3032	Component - No, Yes Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3032	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
Hash DRBG	A3032	Prediction Resistance - Yes Supports Reseed - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Entropy Input - Entropy Input: 128-256 Increment 64, Entropy Input: 192-256 Increment 64, Entropy Input: 256-320 Increment 64 Nonce - Nonce: 128-160 Increment 32, Nonce: 96-128 Increment 32 Personalization String Length - Personalization String Length: 0-256 Increment 128 Additional Input - Additional Input: 0-256 Increment 128 Returned Bits - 160, 224, 256, 384, 512	SP 800-90A Rev. 1
HMAC DRBG	A3032	Prediction Resistance - Yes Supports Reseed - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Entropy Input - Entropy Input: 160-256 Increment 32, Entropy Input: 192-256 Increment 64, Entropy Input: 256-512 Increment 64, Entropy Input: 384-512 Increment 64, Entropy Input: 512-1024 Increment 64 Nonce - Nonce: 128, Nonce: 64, Nonce: 96 Personalization String Length - Personalization String Length: 0-192 Increment 64, Personalization String Length: 0-256 Increment 128 Additional Input - Additional Input: 0-256 Increment 128, Additional Input: 192 Returned Bits - 160, 224, 256, 384, 512	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A3032	MAC - MAC: 32-160 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3032	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3032	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3032	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3032	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3032	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3032	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3032	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3032	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3032	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3032	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A3032	Curve - P-256, P-384, P-521	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A3032	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
KAS-FFC-SSC Sp800-56Ar3	A3032	Domain Parameter Generation Methods - FB, FC, ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-IFC-SSC	A3032	Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Scheme - KAS1 - KAS Role - initiator, responder KAS2 - KAS Role - initiator, responder Fixed Public Exponent - 010001	SP 800-56A Rev. 3
KDA HKDF SP800-56Cr2	A3032	Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Perform Multiple Expansion Tests - No	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A3032	Auxiliary Function Methods - Auxiliary Function Name - SHA2-512 MAC Salting Methods - default, random Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A3032	MAC Salting Methods - default, random Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation KDF Mode - feedback MAC Modes - HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 Fixed Data Order - after fixed data Counter Lengths - 8 The KDF supports an empty IV - Yes The KDF requires an empty IV - Yes Supported Lengths - Supported Lengths: 2048 Derived Key Length - 2048	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
		Shared Secret Length - Shared Secret Length: 224-8192 Increment 8 Perform Multiple Expansion Tests - No	
KDF ANS 9.42 (CVL)	A3032	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Other Info Length - Other Info Length: 0-4096 Increment 8 zz Length - zz Length: 8-4096 Increment 8 Key Data Length - Key Data Length: 8-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-120 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A3032	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Field Size - 224, 571 Shared Info Length - Shared Info Length: 0, 1024 Key Data Length - Key Data Length: 128, 4096	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A3032	Initiator Nonce Length - Initiator Nonce Length: 2048 Responder Nonce Length - Responder Nonce Length: 2048 Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Derived Keying Material Length - Derived Keying Material Length: 3072 Hash Algorithm - SHA-1	SP 800-135 Rev. 1
KDF SNMP (CVL)	A3032	Password Length - Password Length: 256, 64 Engine ID - 000002b87766554433221100, 800002B805123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456	SP 800-135 Rev. 1
KDF SP800-108	A3032	KDF Mode - Counter, Feedback MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 Supported Lengths - Supported Lengths: 8, 72, 128, 776, 3456, 4096 Fixed Data Order - Before Fixed Data Counter Length - 32 Supports Empty IV - No, Yes Custom Key In Length - 0 Requires Empty IV - Yes	SP 800-108 Rev. 1
KDF SRTP (CVL)	A3032	AES Key Length - 128, 192, 256 Supports Empty KDR - No KDR Exponents - 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, 22, 23, 24, 3, 4, 5, 6, 7, 8, 9	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
KDF SSH (CVL)	A3032	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KMAC-128	A3032	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No, Yes	SP 800-185
KMAC-256	A3032	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No, Yes	SP 800-185
KTS-IFC	A3032	Function - keyPairGen, partialVal IUT ID - DEADDEAD Modulo - 2048, 3072, 4096, 6144 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Associated Data Encoding - concatenation Key Length - 1024	SP 800-56B Rev. 2
PBKDF	A3032	Iteration Count - Iteration Count: 1-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 8 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-4)	A3032	Key Generation Mode - B.3.6 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Info Generated By Server - Yes Public Exponent Mode - Random Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3032	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-256	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
RSA Signature Primitive (CVL)	A3032	Private Key Format - crt Public Exponent Mode - fixed Fixed Public Exponent - 010001	FIPS 186-4
RSA SigVer (FIPS186-4)	A3032	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA-1 Public Exponent Mode - Random	FIPS 186-4
Safe Primes Key Generation	A3032	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096	SP 800-56A Rev. 3
Safe Primes Key Verification	A3032	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096	SP 800-56A Rev. 3
SHA-1	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512/224	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512/256	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-224	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-384	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-512	A3032	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHAKE-128	A3032	Supports Bit-Oriented Messages - No Supports Empty Message - Yes	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
		Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	
SHAKE-256	A3032	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TDES-CBC	A3032	Direction - Decrypt Keying Option - 1	SP 800-67 Rev. 2
TDES-CMAC	A3032	Direction - Verification Keying Option - 1 MAC Length - MAC Length: 64 Message Length - Message Length: 0-65536 Increment 8	SP 800-67 Rev. 2
TDES-ECB	A3032	Direction - Decrypt Keying Option - 1	SP 800-67 Rev. 2
TLS v1.2 KDF RFC7627 (CVL)	A3032	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A3032	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 5: Approved Algorithms - PAA

Non-PAA

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A3252	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-512 Increment 8	SP 800-38A
AES-CBC-CS2	A3252	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 128-512 Increment 8	SP 800-38A
AES-CBC-CS3	A3252	Direction - decrypt, encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 136-512 Increment 8	SP 800-38A
AES-CCM	A3252	Key Length - 128, 192, 256 Tag Length - 112, 128, 32, 48, 64, 80, 96 IV Length - IV Length: 56-104 Increment 8	SP 800-38C

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 0-256 Increment 8 AAD Length - AAD Length: 0-524288 Increment 8	
AES-CFB1	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A3252	Direction - Generation, Verification Key Length - 128, 192, 256 MAC Length - MAC Length: 128 Message Length - Message Length: 0-524288 Increment 8	SP 800-38B
AES-CTR	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - No Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3252	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96-1024 Increment 8 Payload Length - Payload Length: 8-65536 Increment 8 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
AES-GMAC	A3252	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
AES-KW	A3252	Direction - Decrypt, Encrypt Cipher - Cipher, Inverse Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F
AES-KWP	A3252	Direction - Decrypt, Encrypt Cipher - Cipher, Inverse Key Length - 128, 192, 256 Payload Length - Payload Length: 8-4096 Increment 8	SP 800-38F
AES-OFB	A3252	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-XTS Testing Revision 2.0	A3252	Direction - Decrypt, Encrypt Key Length - 128, 256 Payload Length - Payload Length: 128-65536 Increment 128 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
Counter DRBG	A3252	Prediction Resistance - Yes Supports Reseed - Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes Additional Input - Additional Input: 0-256 Increment 256, Additional Input: 256, Additional Input: 320, Additional Input: 384 Entropy Input - Entropy Input: 128-256 Increment 128, Entropy Input: 256, Entropy Input: 256-512 Increment 128, Entropy Input: 320, Entropy Input: 384 Nonce - Nonce: 0, Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 256, Personalization String Length: 256, Personalization String Length: 320, Personalization String Length: 384 Returned Bits - 256	SP 800-90A Rev. 1
DSA KeyGen (FIPS186-4)	A3252	L - 2048, 3072 N - 224, 256	FIPS 186-4
DSA PQGGen (FIPS186-4)	A3252	P/Q Generation Methods - Probable G Generation Methods - Canonical, Unverifiable L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA PQGVer (FIPS186-4)	A3252	P/Q Generation Methods - Probable G Generation Methods - Canonical, Unverifiable L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA SigGen (FIPS186-4)	A3252	L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA SigVer (FIPS186-4)	A3252	L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A3252	Curve - P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3252	Curve - P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3252	Component - No, Yes Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3252	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
Hash DRBG	A3252	Prediction Resistance - Yes Supports Reseed - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Entropy Input - Entropy Input: 128-256 Increment 64, Entropy Input: 192-256 Increment 64, Entropy Input: 256-320 Increment 64 Nonce - Nonce: 128-160 Increment 32, Nonce: 96-128 Increment 32 Personalization String Length - Personalization String Length: 0-256 Increment 128 Additional Input - Additional Input: 0-256 Increment 128 Returned Bits - 160, 224, 256, 384, 512	SP 800-90A Rev. 1
HMAC DRBG	A3252	Prediction Resistance - Yes Supports Reseed - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Entropy Input - Entropy Input: 160-256 Increment 32, Entropy Input: 192-256 Increment 64, Entropy Input: 256-512 Increment 64, Entropy Input: 384-512 Increment 64, Entropy Input: 512-1024 Increment 64 Nonce - Nonce: 128, Nonce: 64, Nonce: 96 Personalization String Length - Personalization String Length: 0-192 Increment 64, Personalization String Length: 0-256 Increment 128 Additional Input - Additional Input: 0-256 Increment 128, Additional Input: 192 Returned Bits - 160, 224, 256, 384, 512	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A3252	MAC - MAC: 32-160 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3252	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3252	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3252	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3252	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A3252	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A3252	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A3252	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A3252	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A3252	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A3252	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-524288 Increment 8	FIPS 198-1
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A3252	Curve - P-256, P-384, P-521	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A3252	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
KAS-FFC-SSC Sp800-56Ar3	A3252	Domain Parameter Generation Methods - FB, FC, ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-IFC-SSC	A3252	Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Scheme - KAS1 - KAS Role - initiator, responder KAS2 - KAS Role - initiator, responder Fixed Public Exponent - 010001	SP 800-56A Rev. 3
KDA HKDF SP800-56Cr2	A3252	Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Perform Multiple Expansion Tests - No Uses Hybrid Shared Secret - No	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A3252	Auxiliary Function Methods - Auxiliary Function Name - SHA2-512 MAC Salting Methods - default, random Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A3252	MAC Salting Methods - default, random Fixed Info Pattern - algorithmId||l||uPartyInfo||vPartyInfo Fixed Info Encoding - concatenation KDF Mode - feedback MAC Modes - HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/224, HMAC-SHA2-512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 Fixed Data Order - after fixed data Counter Lengths - 8 The KDF supports an empty IV - Yes The KDF requires an empty IV - Yes Supported Lengths - Supported Lengths: 2048	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
		Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8 Perform Multiple Expansion Tests - No Uses Hybrid Shared Secret - No	
KDF ANS 9.42 (CVL)	A3252	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Other Info Length - Other Info Length: 0-4096 Increment 8 zz Length - zz Length: 8-4096 Increment 8 Key Data Length - Key Data Length: 8-4096 Increment 8 Supplemental Information Length - Supplemental Information Length: 0-120 Increment 8 OID - AES-128-KW, AES-192-KW, AES-256-KW	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A3252	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Field Size - 224, 571 Shared Info Length - Shared Info Length: 0, 1024 Key Data Length - Key Data Length: 128, 4096	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A3252	Initiator Nonce Length - Initiator Nonce Length: 2048 Responder Nonce Length - Responder Nonce Length: 2048 Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Derived Keying Material Length - Derived Keying Material Length: 3072 Hash Algorithm - SHA-1	SP 800-135 Rev. 1
KDF SNMP (CVL)	A3252	Password Length - Password Length: 256, 64 Engine ID - 000002b87766554433221100, 800002B805123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456	SP 800-135 Rev. 1
KDF SP800-108	A3252	KDF Mode - Counter, Feedback MAC Mode - CMAC-AES128, CMAC-AES192, CMAC-AES256, HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 Supported Lengths - Supported Lengths: 8, 72, 128, 776, 3456, 4096 Fixed Data Order - Before Fixed Data Counter Length - 32 Supports Empty IV - No, Yes Custom Key In Length - 0 Requires Empty IV - Yes	SP 800-108 Rev. 1
KDF SRTP (CVL)	A3252	AES Key Length - 128, 192, 256 Supports Empty KDR - No KDR Exponents - 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, 22, 23, 24, 3, 4, 5, 6, 7, 8, 9	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
KDF SSH (CVL)	A3252	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KMAC-128	A3252	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No, Yes	SP 800-185
KMAC-256	A3252	Message Length - Message Length: 0-65536 Increment 8 MAC Length - MAC Length: 32-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8 Hex Customization - No Supports eXtendable-Output Functions - No, Yes	SP 800-185
KTS-IFC	A3252	Function - keyPairGen, partialVal IUT ID - DEADDEAD Modulo - 2048, 3072, 4096, 6144 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Fixed Public Exponent - 010001 Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Hash Algorithms - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Supports Null Associated Data - Yes Associated Data Encoding - concatenation Key Length - 1024	SP 800-56B Rev. 2
PBKDF	A3252	Iteration Count - Iteration Count: 1-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Password Length - Password Length: 8-128 Increment 8 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 112-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-4)	A3252	Key Generation Mode - B.3.6 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Info Generated By Server - Yes Public Exponent Mode - Random Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A3252	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-256	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
RSA Signature Primitive (CVL)	A3252	Private Key Format - crt Public Exponent Mode - fixed Fixed Public Exponent - 010001	FIPS 186-4
RSA SigVer (FIPS186-4)	A3252	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA-1 Public Exponent Mode - Random	FIPS 186-4
Safe Primes Key Generation	A3252	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096	SP 800-56A Rev. 3
Safe Primes Key Verification	A3252	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, modp-2048, modp-3072, modp-4096	SP 800-56A Rev. 3
SHA-1	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-224	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-256	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-384	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512/224	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA2-512/256	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 180-4
SHA3-224	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-256	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-384	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHA3-512	A3252	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
SHAKE-128	A3252	Supports Bit-Oriented Messages - No Supports Empty Message - Yes	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
		Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	
SHAKE-256	A3252	Supports Bit-Oriented Messages - No Supports Empty Message - Yes Supports Bit-Oriented Output - No Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TDES-CBC	A3252	Direction - Decrypt Keying Option - 1	SP 800-67 Rev. 2
TDES-CMAC	A3252	Direction - Verification Keying Option - 1 MAC Length - MAC Length: 64 Message Length - Message Length: 0-65536 Increment 8	SP 800-67 Rev. 2
TDES-ECB	A3252	Direction - Decrypt Keying Option - 1	SP 800-67 Rev. 2
TLS v1.2 KDF RFC7627 (CVL)	A3252	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A3252	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms - Non-PAA

The module implements the cryptographic algorithms listed in the above tables. The module also supports RSA KeyGen, SigGen and SigVer with modulus size greater than 4096, where CAVP testing is not available.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG1	Key Type:Symmetric and Asymmetric	N/A	Sections 4 example 1, 5.1, 5.2 and 6.2 of NIST SP 800-133 rev2
CKG2	Key Type:Symmetric	N/A	Section 6.3 of NIST SP 800-133 rev2

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

There are no non-approved and allowed algorithms, hence the table is excluded.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

There are no algorithms that are non-approved but allowed with no security claimed, hence the table is excluded.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
EdDSA KeyGen	Asymmetric Key Generation (Ed25519, Ed448, X25519, and X448)
EdDSA SigGen	Signature generation using Edwards curves (ED25519, ED448)
EdDSA SigVer	Signature verification using Edwards curves (ED25519, ED448)
RSA Primitives	RSA Signature generation, verification, encrypt and decrypt primitives

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Random Number Generation	DRBG	Used for random number and symmetric key generation		Counter DRBG: (A3032, A3252) Hash DRBG: (A3032, A3252) HMAC DRBG: (A3032, A3252)
Asymmetric Key Generation	AsymKeyPair-KeyGen CKG	Used to generate DSA, ECDSA, RSA, DH, ECDH, keys		DSA KeyGen (FIPS186-4): (A3032, A3252) ECDSA KeyGen (FIPS186-4): (A3032, A3252) RSA KeyGen (FIPS186-4): (A3032, A3252) DSA PQGGen

Name	Type	Description	Properties	Algorithms
				(FIPS186-4): (A3032, A3252) Safe Primes Key Generation: (A3032, A3252) CKG1: ()
Key Derivation Function (KDF)	KAS-135KDF KAS-56CKDF KBKDF PBKDF	Used to derive keys using KBKDF, PBKDF2, HKDF, SP 800-56C rev2, One-Step KDF (KDA), Two-Step KDF (KDA), SP 800-135 rev1 TLS 1.2, SSHv2, SNMPv3, SRTP, IKEv2, ANSI X9.63-2001, ANSI X9.42-2001 KDFs and TLS 1.3 KDF		KDA HKDF SP800-56Cr2: (A3032, A3252) KDF ANS 9.42: (A3032, A3252) KDF ANS 9.63: (A3032, A3252) KDF IKEv2: (A3032, A3252) KDF SNMP: (A3032, A3252) KDF SP800-108: (A3032, A3252) KDF SRTP: (A3032, A3252) KDF SSH: (A3032, A3252) PBKDF: (A3032, A3252) TLS v1.2 KDF RFC7627: (A3032, A3252) TLS v1.3 KDF: (A3032, A3252) KDA OneStep SP800-56Cr2: (A3032,

Name	Type	Description	Properties	Algorithms
				A3252) KDA TwoStep SP800-56Cr2: (A3032, A3252)
Symmetric Encrypt/Decrypt	BC-Auth BC-UnAuth	Used to encrypt or decrypt data. TDES: decrypt only. Executes using AES EDK/TDES DK (passed in by the calling application)		AES-CBC: (A3032, A3252) AES-CBC-CS1: (A3032, A3252) AES-CBC-CS2: (A3032, A3252) AES-CBC-CS3: (A3032, A3252) AES-CCM: (A3032, A3252) AES-CFB1: (A3032, A3252) AES-CFB128: (A3032, A3252) AES-CFB8: (A3032, A3252) AES-CTR: (A3032, A3252) AES-ECB: (A3032, A3252) AES-GCM: (A3032, A3252) AES-GMAC: (A3032, A3252) AES-OFB: (A3032, A3252) AES-XTS Testing Revision 2.0:

Name	Type	Description	Properties	Algorithms
				(A3032, A3252) TDES-CBC: (A3032, A3252) TDES-ECB: (A3032, A3252)
Message Digest (SHS)	SHA	Used to generate a SHA-1, SHA-2, or SHA-3 message digest		SHA-1: (A3032, A3252) SHA2-224: (A3032, A3252) SHA2-256: (A3032, A3252) SHA2-384: (A3032, A3252) SHA2-512: (A3032, A3252) SHA2-512/224: (A3032, A3252) SHA2-512/256: (A3032, A3252) SHA3-224: (A3032, A3252) SHA3-256: (A3032, A3252) SHA3-384: (A3032, A3252) SHA3-512: (A3032, A3252) SHAKE-128: (A3032, A3252) SHAKE-256: (A3032, A3252)

Name	Type	Description	Properties	Algorithms
Keyed Hash (HMAC/KMAC/CMAC)	MAC	Used to generate or verify data integrity with HMAC, KMAC or CMAC. TDES: verify only. Executes using HMAC , KMAC, AES or TDES Key (passed in by the calling application)		HMAC-SHA-1: (A3032, A3252) HMAC-SHA2-224: (A3032, A3252) HMAC-SHA2-256: (A3032, A3252) HMAC-SHA2-384: (A3032, A3252) HMAC-SHA2-512: (A3032, A3252) HMAC-SHA2-512/224: (A3032, A3252) HMAC-SHA2-512/256: (A3032, A3252) HMAC-SHA3-224: (A3032, A3252) HMAC-SHA3-256: (A3032, A3252) HMAC-SHA3-384: (A3032, A3252) HMAC-SHA3-512: (A3032, A3252) KMAC-128: (A3032, A3252) KMAC-256: (A3032, A3252) AES-CMAC: (A3032, A3252) TDES-CMAC: (A3032, A3252)

Name	Type	Description	Properties	Algorithms
Key Wrapping (KW)	BC-UnAuth	Used to encrypt a key value on behalf of the calling application. Executes using AES Key Wrapping Key (passed in by the calling application). Key sizes 128, 192 and 256 providing 128, 192 and 256 bits of encryption strength. AES-KW, AES-KWP is CAVP tested per FIPS 140-3 IG D.G.		AES-KW: (A3032, A3252) AES-KWP: (A3032, A3252)
Digital Signature	DigSig-SigGen DigSig-SigVer	Used to generate or verify RSA, DSA, ECDSA, digital signatures. Executes using RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK, (passed in by the calling application)		DSA SigGen (FIPS186-4): (A3032, A3252) DSA SigVer (FIPS186-4): (A3032, A3252) ECDSA SigGen (FIPS186-4): (A3032, A3252) ECDSA SigVer (FIPS186-4): (A3032, A3252) RSA SigGen (FIPS186-4): (A3032, A3252) DSA PQGVer (FIPS186-4): (A3032, A3252)

Name	Type	Description	Properties	Algorithms
				RSA Signature Primitive: (A3032, A3252) RSA SigVer (FIPS186-4): (A3032, A3252)
Asymmetric Key Verification	AsymKeyPair-KeyVer	Used to verify ECDSA public key and SafePrime keys		ECDSA KeyVer (FIPS186-4): (A3032, A3252) Safe Primes Key Verification: (A3032, A3252)
Shared Secret Computation	KAS-SSC	Used to perform key agreement primitives on behalf of the calling application (does not establish keys into the module). Executes using DH Private, DH Public, EC DH Private, EC DH Public, RSA SGK, RSA SVK (passed in by the calling application). For ECC: Curves P-256, P-384 and P-521 providing 128 to 256 bits of encryption strength. For FFC: 2048, 3072 and 4096		KAS-ECC CDH-Component SP800-56Ar3: (A3032, A3252) KAS-ECC-SSC Sp800-56Ar3: (A3032, A3252) KAS-FFC-SSC Sp800-56Ar3: (A3032, A3252) KAS-IFC-SSC: (A3032, A3252)

Name	Type	Description	Properties	Algorithms
		bit keys providing 112 to 152 bits of security strength. For IFC: 2048, 3072, 4096, 6144, 8192 bit modulus providing 112 to 200 bits of encryption strength. The module follows SP 800-56A rev3 KAS-ECC-SSC (FIPS 140-3 IG D.F Scenario 2 path 1), SP 800-56A rev3 KAS-FFC-SSC (FIPS 140-3 IG D.F Scenario 2 path 1) and SP 800-56B rev2 KAS-IFC-SSC (FIPS 140-3 IG D.F Scenario 1 path 1)		
Key Transport Primitive	KTS-Decap KTS-Encap	Used for key transport, supports KTS-OAEP. 2048, 3072, 4096 and 6144 bit modulus providing 112 to 176 bits of encryption strength. The module follows SP 800-56B rev2 KTS-IFC (FIPS 140-3 IG D.G). Does not establish keys		KTS-IFC: (A3032, A3252)

Name	Type	Description	Properties	Algorithms
		into the module.		
Symmetric Key Generation	CKG	Used to generate symmetric keys per NIST SP 800-133 rev2		CKG2: ()

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

AES GCM IV Generation

In the case of AES-GCM, the IV generation method is user-selectable, and the value can be computed in more than one manner as follows:

1) TLS 1.2: The module's AES-GCM implementation conforms to IG C.H, scenario #1, following RFC 5288. The module is compatible with TLS 1.2 protocol and provides the primitives to support the AES GCM cipher suites from SP 800-52 rev1 Section 3.3.1. The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key in accordance with RFC 5246 for TLS 1.2, respectively.

2) IKEv2: The module's AES-GCM implementation conforms to IG C.H, scenario #1 following RFC 7296 for IPSec/IKEv2. The AES GCM IV is generated according to RFC5282. The counter portion of the IV is set by the module within its cryptographic boundary. When the IV exhausts the maximum number of possible values for a given session key, the first party, client or server, to encounter this condition will trigger a handshake to establish a new encryption key. In case the module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption shall be established.

3) TLS 1.3: The module's AES-GCM implementation conforms to IG C.H Scenario#5. The module is compatible with TLS v1.3 and provides support for the acceptable GCM cipher suites from Section 8.4 of RFC 8446 and confirms that the IV is generated and used within the protocol's implementation. The counter portion of the IV is set by the module within its cryptographic boundary. In case the module's power is lost and then restored, a new key for use with the AES GCM encryption/decryption must be established.

4) Non-protocol specific usage: The module's AES-GCM implementation conforms to IG C.H, scenario #3, when operating in approved mode of operation, AES GCM, IVs are generated both internally and deterministically and are a minimum of 96-bits in length as specified in SP 800-38D, Section 8.2.1. The selection of the IV construction method is the responsibility of the user of this cryptographic module.

Note: Externally generated IVs are not allowed for AES-GCM encryption.

PBKDF

In line with the requirements of SP 800-132 and FIPS 140-3 IG D.N, keys generated using the approved PBKDF must only be used for storage applications. The algorithm uses option 1a as specified in SP 800-132 Section 5.4. Any other use of the approved PBKDF is non-conformant. In approved mode the module enforces that any password used must encode to at least 14 bytes (112 bits) and that the salt is at least 16 bytes (128 bits) long. The iteration count associated with the PBKDF should be as large as practical.

As the module is a general-purpose firmware module, it is not possible to anticipate all the levels of use for the PBKDF, however a user of the module should also note that a password should at least contain enough entropy to be unguessable and contain enough entropy to reflect the security strength required for the key being generated.

AES-XTS

In line with the requirements of SP 800-38E and FIPS 140-3 IG C.I, the keys are generated independently according to Section 6.3 of SP 800-133 rev2 and verification of the keys (key1 ≠ key2) is performed before using them in the AES-XTS algorithm.

Key Agreement

The module implements the following CAVP tested key agreement methods:
SP 800-56A rev3 KAS-ECC-SSC (FIPS 140-3 IG D.F Scenario 2 path 1)
SP 800-56A rev3 KAS-FFC-SSC (FIPS 140-3 IG D.F Scenario 2 path 1)
SP 800-56B rev2 KAS-IFC-SSC (FIPS 140-3 IG D.F Scenario 1 path 1)

SHA3 and SHAKE

Per FIPS 140-3 IG C.C, all SHA3 and SHAKE functions are tested on all the operational environments. The higher-level algorithms using SHA3 (HMAC-SHA3) are also tested on all the operational environments.

RSA

Per FIPS 140-3 IG C.F, RSA SigGen is tested with 2048, 3072, 4096-bit modulus and RSA SigVer is tested with 1024, 2048, 3072, 4096-bit modulus. The module also supports RSA KeyGen, SigGen and SigVer with modulus size greater than 4096, for which CAVP testing is not available.

Legacy use algorithms

Per SP 800-131A rev2, TDES-CBC/TDES-ECB Decrypt, TDES-CMAC Verification, DSA/ECDSA/RSA SigVer using SHA-1 is allowed for legacy use. The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2031.

2.8 RBG and Entropy

N/A for this module.

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
N/A	Non-Physical	N/A	N/A	N/A	

Table 10: Entropy Sources

N/A for this module. The module passively receives entropy from outside the boundary. The caveat “No assurance of the minimum strength of generated SSPs (e.g., keys)” applies to this module.

Applications shall use entropy sources that meet the security strength required for the random number generation mechanism as shown in [SP 800-90A rev1] Table 2 (Hash_DRBG, HMAC_DRBG, CTR_DRBG). A minimum of 112-bits of entropy must be supplied. This entropy is supplied by means of callback functions. Those functions must return an error if the minimum entropy strength cannot be met.

2.9 Key Generation

The module generates symmetric and asymmetric keys following the sections of SP 800-133 rev2 as specified in Table “Vendor-Affirmed Algorithms” above.

Private and secret keys as well as seeds and entropy input are provided to the module by the calling application and are destroyed when released by the appropriate API function calls. Keys residing in internally allocated data structures (during the lifetime of an API call) can only be accessed using the module defined API. The operating system protects application space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions (Module Services) are executed by the calling application invoking an API. Each API either succeeds or fails and is logically non-interruptible from the point of view of the calling application.

The module supports generation of ECDSA, RSA, DSA, EC Diffie-Hellman and Diffie-Hellman key pairs per Section 5 in SP 800-133 rev2. The output of SP 800-90A rev1 random bit generator is used for generating the seed used in asymmetric key generation. The module also complies with Sections 6.1 and 6.2 of SP 800-133 rev2.

2.10 Key Establishment

The module implements key agreement methods per FIPS 140-3 IG D.F and key transport methods per FIPS 140-3 IG D.G (SP 800-38F AES-KW and AES-KWP, SP 800-56B rev2 KTS-IFC). Detailed information is provided in Table “Security Function Implementations” Section above.

2.11 Industry Protocols

In reference to FIPS 140-3 IG D.C, the module implements the KDFs of SSH, TLS, IKE, SRTP, SNMP, ANS X9.42 and ANS X9.63 but no parts of the protocols other than the approved cryptographic algorithms and the KDFs have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API entry point data input stack parameters
N/A	Data Output	API output parameters resulting from call execution
N/A	Control Input	API entry point and corresponding stack parameters
N/A	Status Output	API return value resulting from call execution

Table 11: Ports and Interfaces

The logical interface is a C-language application program interface (API). The Data Input interface consists of the input parameters of the API functions. The Data Output interface consists of the output parameters of the API functions. The Control Input interface consists of the actual API functions. The Status Output interface includes the return values of the API functions.

3.2 Control Interface Not Inhibited

Please note that the module does not support a control output interface and is not applicable for this module.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not implement authentication mechanisms and does not allow concurrent operators.

4.2 Roles

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

Table 12: Roles

The module meets all FIPS 140-3 level 1 requirements for Roles. The Module implements both a User Role (User) as well as the Crypto Officer (CO) role. The User and Crypto Officer roles are implicitly assumed by the application accessing services implemented by the Module.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Random Number Generation	Used for random number and symmetric key generation	1	DRBG struct (RBG State); DRBG_Seed	Status return; Random value	Random Number Generation Symmetric Key Generation	Crypto-Officer - DRBG_C: W,E - Entropy Input: W,E,Z - DRBG_Key: W,E - DRBG_Seed: G,E,Z - DRBG_V: W,E
Asymmetric Key Generation	Generate asymmetric key pairs	1	ECDSA: curve identifier. DSA, RSA: domain parameter targets	Status return; general digital signature private and public keys	Asymmetric Key Generation	Crypto-Officer - RSA SGK: G,R - ECDSA SGK: G,R - DSA SGK: G,R - RSA SVK: G,R - ECDSA SVK: G,R - DSA SVK: G,R - RSA KDK: G,R - RSA KEK: G,R
Key Derivation	Used to derive	1	Key agreemen	Status return;	Key Derivation Function (KDF)	Crypto-Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Function (KDF)	keys using KBKDF, PBKDF2, HKDF, SP 800-56C rev2 One-Step KDF (KDA), SP 800-56C rev2 Two-Step KDF (KDA), SP 800-135 rev1 TLS 1.2, SSHv2, SNMPv3, SRTP, IKEv2, ANSI X9.6-2001, ANSI X9.42-2001 KDFs and TLS 1.3 KDF		t shared secret; flags	derived keying material		- KDF Derived Key: G,R
Symmetric Encrypt/Decrypt	Used to encrypt or decrypt data. Executes using AES EDK (passed in by the calling application)	1	Encryption or decryption key; plaintext or ciphertext data; flags	Status return. Plaintext or ciphertext data	Symmetric Encrypt/Decrypt	Crypto-Officer - AES EDK: W,E - AES GCM: W,E - AES XTS: W,E - AES Key Wrapping: W,E - TDES DK: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Digest (SHS)	Used to generate a SHA-1, SHA-2, or SHA-3 message digest	1	Data to be hashed	Status return. Hashed data	Message Digest (SHS)	Crypto-Officer
Keyed Hash	Used to generate or verify data integrity with HMAC, KMAC or CMAC. Executes using HMAC, KMAC or AES Key (passed in by the calling application)	1	Data to be hashed and keying material	Status return; MAC output value.	Keyed Hash (HMAC/KMAC/CMAC)	Crypto-Officer - HMAC Key: W,E - KMAC Key: W,E - AES CMAC: W,E - TDES CMAC: W,E
Key Wrapping (KW)	Used to encrypt a key value on behalf of the calling application. Executes using AES Key Wrapping Key (passed in by the calling application). AES-KW, AES-	1	Keying material	Encrypted key	Key Wrapping (KW)	Crypto-Officer - AES Key Wrapping: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	KWP is CAVP tested per FIPS 140-3 IG D.G.					
Key Agreement/ Agreement Component (SP 800-56A rev3)	Used to perform key agreement primitives on behalf of the calling application (does not establish keys into the module). Executes using DH Private, DH Public, EC DH Private, EC DH Public, RSA SGK, RSA SVK (passed in by the calling application)	1	Key structs (key agreement keys); flags	Status return; key agreement shared secret	Shared Secret Computation	Crypto-Officer - DH Private: W,E - EC DH Private: W,E - RSA SGK: W,E - DH Public: W,E - EC DH Public: W,E - RSA SVK: W,E
Digital Signature	Used to generate or verify RSA, DSA, ECDSA,	1	Sign: signing key; message. Verify: signature	Status return; Signature value	Digital Signature	Crypto-Officer - RSA SGK: W,E - RSA SVK: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	digital signatures. Executes using RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK, (passed in by the calling application)		value; flags; sizes			- DSA SGK: W,E - DSA SVK: W,E - ECDSA SGK: W,E - ECDSA SVK: W,E
Asymmetric Key Verification	Used to verify ECDSA keys	1	Public Key	Status return	Asymmetric Key Verification	Crypto-Officer - ECDSA SVK: W,E
Module initialization	The module is initialized when the provider is loaded	1	N/A	N/A	None	Crypto-Officer
Perform Self-Test	Perform self-tests on demand	1	N/A	Success/failure message	None	Crypto-Officer
Key Transport	Used for Key Transport	1	Key to be transported	Encrypted key	Key Transport Primitive	Crypto-Officer - RSA KDK: W,E - RSA KEK: W,E
Show Module Name and Version	Used to output module name	N/A	N/A	name: CiscoSSL FIPS	None	Crypto-Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	and version			Provider version: 8.0		
Show Status	Used to output module status	N/A	N/A	status: active	None	Crypto-Officer
Zeroize	Zeroize SSPs	N/A	Any SSP	N/A	None	Crypto-Officer - RSA SGK: Z - RSA KDK: Z - DSA SGK: Z - ECDSA SGK: Z - DH Private: Z - EC DH Private: Z - AES EDK: Z - AES CMAC: Z - AES GCM: Z - AES XTS: Z - AES Key Wrapping: Z - HMAC Key: Z - DRBG_C: Z - DRBG_Key: Z - DRBG_Seed: Z - DRBG_V: Z - RSA

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						SVK: Z - RSA KEK: Z - DSA SVK: Z - ECDSA SVK: Z - DH Public: Z - EC DH Public: Z - KDF Derived Key: Z - TDES DK: Z - TDES CMAC: Z

Table 13: Approved Services

The module meets all FIPS 140-3 level 1 requirements for Services. The initialization process is described in the Secure Distribution, Operation, and User Guidance section of this document. CO services with associated input and output are listed in the above table. All the services provided by the module can be accessed by both the User and the Crypto Officer roles. The User Role (User) can load the module and call any of the API functions. The Crypto Officer Role (CO) is responsible for installation of the module on the host computer system and calling of any API functions.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Edwards curves Key Generation	Key pair generation using Edwards curves (ED25519, ED448, X25519, X448)	EdDSA KeyGen	Crypto-Officer, User
Edwards curves Digital Signature Generation	Signature generation using Edwards curves (ED25519, ED448)	EdDSA SigGen	Crypto-Officer, User
Edwards curves Digital Signature Verification	Signature verification using Edwards curves (ED25519, ED448)	EdDSA SigVer	Crypto-Officer, User
RSA Primitives	RSA Sign, verify, encrypt, decrypt without hashing/padding	RSA Primitives	Crypto-Officer, User

Table 14: Non-Approved Services

The module implements non-approved services mentioned in the above table. For these specific services, the service indicators ‘3, ED25519, ED448, X25519, X448’ indicates that the service is non-approved.

4.5 External Software/Firmware Loaded

Not Applicable for this module.

5 Software/Firmware Security

5.1 Integrity Techniques

The module runs a HMAC SHA2-256 integrity verification on the shared object file (fips.so) during initialization by the host application. The module also runs the self-test for HMAC SHA2-256 prior to running the integrity check.

5.2 Initiate on Demand

The operator can initiate on-demand integrity test by calling SELF_TEST_post() or rebooting the host platform.

5.3 Additional Information

The public verification key used for firmware integrity test is not an SSP.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

How Requirements are Satisfied:

The module was tested on the platforms listed in Table 2 for the purposes of this FIPS 140-3 validation. The module is expected to execute correctly on any production grade CPU with commonly used operating system. No operational environment restrictions are required for operation in the approved mode.

CiscoSSL FIPS Provider is a Firmware module and classified as a non-modifiable OE. The requirements under ISO/IEC 19790, section 7.6 “Operational environment”, are met by the module for Level 1 firmware requirements.

7 Physical Security

Not Applicable for this module.

8 Non-Invasive Security

Not Applicable for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Volatile Memory	Dynamic

Table 15: Storage Areas

The module stores DRBG state values for the lifetime of the DRBG instance. The module uses CSPs passed in by the calling application on the stack. The module does not store any CSP persistently (beyond the lifetime of an API call), except for DRBG state values used for the module's default key generation service. The module implements SP 800-90A rev1 compliant DRBG services for creation of symmetric keys, and for generation of DSA, elliptic curve, and RSA keys as shown in Table 4. The calling application is responsible for storage of generated keys returned by the module.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Input	Calling process	API input parameters	Plaintext	Manual	Electronic	
API Output	API output parameters	Calling process	Plaintext	Manual	Electronic	

Table 16: SSP Input-Output Methods

All CSPs enter the module's boundary in plaintext as API parameters, associated by memory location. However, none crosses the physical parameter. The module does not output CSPs, other than as explicit results of key generation services or keys passed into the module by the calling application.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
OPENSSL_cleanse()	API call clears the temporarily stored CSPs	Zeroized SSPs will no longer be accessible through API calls	Allowed
Power Cycle	Power Cycle zeroizes all stored SSPs	Operating System zeroizes all the stored SSPs	Allowed

Table 17: SSP Zeroization Methods

Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. The calling application is responsible for parameters passed in and out of the module. Successful completion of the zeroization service is determined by clean execution of OPENSSL_cleanse() without any errors being returned or a successful reboot of the host platform.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA SGK	Used to generate Digital Signatures	2048, 3072, 4096 bits - 112, 128, 152 bits	Signature Generation Key - CSP	Asymmetric Key Generation		Digital Signature
RSA KDK	Used in Asymmetric Key Operation to to decrypt keys	2048, 3072 4096 bits - 112, 128, 152 bits	Key Transport Key - CSP	Asymmetric Key Generation		Key Transport Primitive
DSA SGK	Used to generate Digital Signatures	2048, 3072 bits - 112, 128 bits	Signature Generation Key - CSP	Asymmetric Key Generation		Digital Signature
ECDSA SGK	Used to generate Digital Signatures	256, 384, 521 bits - 128, 192, 256 bits	Signature Generation Key - CSP	Asymmetric Key Generation		Digital Signature
DH Private	Used for Key Agreement	2048, 3072 bits - 112, 128 bits	Key Agreement Key - CSP	Asymmetric Key Generation		Shared Secret Computation
EC DH Private	Used for Key Agreement	256, 384, 521 bits - 128, 192, 256 bits	Key Agreement Key - CSP	Asymmetric Key Generation		Shared Secret Computation
AES EDK	Used for Symmetric encrypt and	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP	Random Number Generation		Symmetric Encrypt/Decrypt

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	decrypt operations					
AES CMAC	Used for MAC calculation and verification	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP	Random Number Generation		Keyed Hash (HMAC/KMAC/CMAC)
AES GCM	Used for authenticated cipher operations	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP	Random Number Generation		Symmetric Encrypt/Decrypt
AES XTS	Used for cipher operation	128, 256 bits - 128, 256 bits	Symmetric Key - CSP	Random Number Generation		Symmetric Encrypt/Decrypt
AES Key Wrapping	Used for key wrapping	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP	Random Number Generation		Key Wrapping (KW)
HMAC Key	Used for MAC generation and verification	128 to 524288 bits - greater than 128 bits	Keyed Hash - CSP	Random Number Generation		Keyed Hash (HMAC/KMAC/CMAC)
KMAC Key	Used for MAC generation	256, 512 bits - 128, 256 bits	Keyed Hash - CSP			Keyed Hash (HMAC/KMAC/CMAC)
DRBG_C	Element of Hash DRBG state, defined per FIPS 140-3 IG D.L	440-888 bits - 160-256 bits	DRBG State - CSP	Random Number Generation		Random Number Generation
Entropy Input	Entropy input from an external source	128-2^35 - 128 - 256	Entropy Input - CSP			Random Number Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	used for DRBG seeding, defined per FIPS 140-3 IG D.L					
DRBG_Key	Element of CTR DRBG or HMAC DRBG state, defined per FIPS 140-3 IG D.L	CTR_DRBG: 128-256, HMAC DRBG: 128-256 - CTR_DRBG: 128 - 256, HMAC DRBG: 160 - 256	CTR_DRBG_Key, HMAC_DRBG_Key - CSP	Random Number Generation		Random Number Generation
DRBG_Seed	Seed used for DRBG Instantiation and Reseed, defined per FIPS 140-3 IG D.L	128-256 - 128 - 256	DRBG Seed - CSP	Random Number Generation		Random Number Generation
DRBG_V	Element of CTR, Hash or HMAC DRBG state, defined per FIPS 140-3 IG D.L	CTR_DRBG: 128-256, Hash DRBG: 128-256, HMAC DRBG: 128-256 - CTR_DRBG: 128 - 256, Hash DRBG: 128 - 256, HMAC	DRBG State - CSP	Random Number Generation		Random Number Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		DRBG: 128 - 256				
RSA SVK	RSA signature verification public key	1024, 2048, 3072, 4096 bits - 80, 112, 128, 152 bits	Verification Key - PSP	Asymmetric Key Generation		Digital Signature
RSA KEK	RSA key encryption (public key transport) key	2048, 3072, 4096 bits - 112, 128, 152 bits	Encryption Key - PSP	Asymmetric Key Generation		Key Transport Primitive
DSA SVK	DSA signature verification key	1024, 2048, 3072 bits - 80, 112, 128 bits	Verification Key - PSP	Asymmetric Key Generation		Digital Signature
ECDSA SVK	ECDSA signature verification key	233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - 112, 128, 192, 256 bits	Verification Key - PSP	Asymmetric Key Generation		Digital Signature
DH Public	DH public key agreement key	2048, 3072 bits - 112, 128 bits	Public Key Agreement Key - PSP	Asymmetric Key Generation		Shared Secret Computation
EC DH Public	EC DH public key agreement key	256, 384, 521 bits - 128, 192, 256 bits	Public Key Agreement Key - PSP	Asymmetric Key Generation		Shared Secret Computation
KDF Derived Key	Key derived from KDFs	128, 256 bits - 128, 256 bits	Derived Key - CSP	Key Derivation Function (KDF)		Key Derivation Function (KDF)
TDES DK	TDES Decryption key	168 bits - 112 bits	Symmetric Key - CSP	Random Number Generation		Symmetric Encrypt/Decrypt

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TDES CMAC	Used for MAC verification	168 bits - 112 bits	Verification Key - PSP	Random Number Generation		Keyed Hash (HMAC/KMAC/CMAC)

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
RSA SGK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	RSA SVK:Paired With
RSA KDK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	RSA KEK:Paired With
DSA SGK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DSA SVK:Paired With
ECDSA SGK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	ECDSA SVK:Paired With
DH Private	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DH Public:Paired With
EC DH Private	API Input API Output	RAM:Plaintext	Until zeroized by reboot	OPENSSL_cleanse() Power Cycle	EC DH Public:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			or API call		
AES EDK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
AES CMAC	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
AES GCM	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
AES XTS	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
AES Key Wrapping	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
HMAC Key	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
KMAC Key	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	

Name	Input-Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG_C	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DRBG_Seed:Derived From DRBG_V:Used With
Entropy Input	API Input	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DRBG_Seed:Constituent
DRBG_Key	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DRBG_Seed:Derived From DRBG_V:Used With
DRBG_Seed		RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DRBG_C:Derives DRBG_Key:Derives DRBG_V:Derives Entropy Input:Incorporates
DRBG_V	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DRBG_Seed:Derived From DRBG_Key:Used With
RSA SVK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	RSA SGK:Paired With
RSA KEK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	RSA KDK:Paired With

Name	Input-Output	Storage	Storage Duration	Zeroization	Related SSPs
DSA SVK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DSA SGK:Paired With
ECDSA SVK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	ECDSA SGK:Paired With
DH Public	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	DH Private:Paired With
EC DH Public	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	EC DH Private:Paired With
KDF Derived Key	API Output		Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
TDES DK	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	
TDES CMAC	API Input API Output	RAM:Plaintext	Until zeroized by reboot or API call	OPENSSL_cleanse() Power Cycle	

Table 19: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3032)	256 bits	Firmware Integrity Test	SW/FW Integrity	Returns 1 when power up self tests succeed	The SELF_TEST_post() function performs all power-up self-tests listed above with no operator intervention required when the module loads, returning a “1” if all power-up self-tests succeed, and a “0” otherwise. The power-up self-tests may also be performed on-demand by calling this function and interpretation of the return code is the responsibility of the calling application
HMAC-SHA2-256 (A3252)	256 bits	Firmware Integrity Test	SW/FW Integrity	Returns 1 when power up self tests succeed	The SELF_TEST_post() function performs all power-up self-tests listed above with no operator intervention required when the module loads, returning a “1” if all power-up self-tests succeed, and a “0” otherwise. The power-up self-tests may also be performed on-demand by calling this function and interpretation of the return code is the responsibility of the calling application

Table 20: Pre-Operational Self-Tests

The module performs firmware integrity test and conditional Cryptographic Algorithm Self-Tests (CASTs) before it is operational. The module is single threaded and will not return to the calling application until the CASTs are complete. If the self-tests fail, the module goes to an error state and subsequent calls to the module will fail and thus no further cryptographic operations are possible. The CO can clear the error state by restarting the host platform.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB Encrypt (A3032)	128 bits	KAT	CAST	Returns 1 on successful completion	Encrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-ECB Encrypt (A3252)	128 bits	KAT	CAST	Returns 1 on successful completion	Encrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-GCM Encrypt (A3032)	256 bits	KAT	CAST	Returns 1 on successful completion	Encrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-GCM Encrypt (A3252)	256 bits	KAT	CAST	Returns 1 on successful completion	Encrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-CMAC Generate (A3032)	128, 192, 256 bits	KAT	CAST	Returns 1 on successful completion	Generate KAT	Upon power-up and call of SELF_TEST_post() function
AES-CMAC Generate (A3252)	128, 192, 256 bits	KAT	CAST	Returns 1 on successful completion	Generate KAT	Upon power-up and call of SELF_TEST_post() function
Counter DRBG (A3032)	AES-128 with derivation function	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function
Counter DRBG (A3252)	AES-128 with derivation function	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function
Hash DRBG (A3032)	SHA2-256	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A3252)	SHA2-256	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function
HMAC DRBG (A3032)	SHA-1	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function
HMAC DRBG (A3252)	SHA-1	KAT	CAST	Returns 1 on successful completion	Instantiate, Generate, Reseed	Upon power-up and call of SELF_TEST_post() function
DSA SigGen (FIPS186-4) (A3032)	2048-bit with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function
DSA SigGen (FIPS186-4) (A3252)	2048-bit with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function
DSA SigVer (FIPS186-4) (A3032)	2048-bit with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
DSA SigVer (FIPS186-4) (A3252)	2048-bit with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
ECDSA SigGen (FIPS186-4) (A3032)	P-256 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function
ECDSA SigGen (FIPS186-4) (A3252)	P-256 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigVer (FIPS186-4) (A3032)	P-256 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
ECDSA SigVer (FIPS186-4) (A3252)	P-256 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
RSA SigGen (FIPS186-4) (A3032)	k=2048 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function
RSA SigGen (FIPS186-4) (A3252)	k=2048 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Sign	Upon power-up and call of SELF_TEST_post() function
RSA SigVer (FIPS186-4) (A3032)	k=2048 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
RSA SigVer (FIPS186-4) (A3252)	k=2048 with SHA2-256	KAT	CAST	Returns 1 on successful completion	Verify	Upon power-up and call of SELF_TEST_post() function
KAS-FFC-SSC Sp800-56Ar3 (A3032)	L=2048/N=256	KAT	CAST	Returns 1 on successful completion	dhEphem Shared Secret (Z) Computation	Upon power-up and call of SELF_TEST_post() function
KAS-FFC-SSC Sp800-56Ar3 (A3252)	L=2048/N=256	KAT	CAST	Returns 1 on successful completion	dhEphem Shared Secret (Z) Computation	Upon power-up and call of SELF_TEST_post() function
KAS-ECC-SSC Sp800-56Ar3 (A3032)	P-256	KAT	CAST	Returns 1 on successful completion	Ephemeral Unified Shared Secret (Z)	Upon power-up and call of SELF_TEST_post() function

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
					Computation	
KAS-ECC-SSC Sp800-56Ar3 (A3252)	P-256	KAT	CAST	Returns 1 on successful completion	Ephemeral Unified Shared Secret (Z) Computation	Upon power-up and call of SELF_TEST_post() function
KAS-IFC-SSC (A3032)	k=2048	KAT	CAST	Returns 1 on successful completion	[SP 800-56B rev2] Section 8.2.2 RSA Primitive Computation	Upon power-up and call of SELF_TEST_post() function
KAS-IFC-SSC (A3252)	k=2048	KAT	CAST	Returns 1 on successful completion	[SP 800-56B rev2] Section 8.2.2 RSA Primitive Computation	Upon power-up and call of SELF_TEST_post() function
SHA-1 (A3032)	SHA-1	KAT	CAST	Returns 1 on successful completion	Simple SHA KAT	Upon power-up and call of SELF_TEST_post() function
SHA-1 (A3252)	SHA-1	KAT	CAST	Returns 1 on successful completion	Simple SHA KAT	Upon power-up and call of SELF_TEST_post() function
SHA2-512 (A3032)	SHA2-512	KAT	CAST	Returns 1 on successful completion	Simple SHA KAT	Upon power-up and call of SELF_TEST_post() function
SHA2-512 (A3252)	SHA2-512	KAT	CAST	Returns 1 on successful completion	Simple SHA KAT	Upon power-up and call of SELF_TEST_post() function
SHA3-256 (A3032)	SHA3-256	KAT	CAST	Returns 1 on successful	Simple SHA KAT	Upon power-up and call of

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				completion		SELF_TEST_post() function
SHA3-256 (A3252)	SHA3-256	KAT	CAST	Returns 1 on successful completion	Simple SHA KAT	Upon power-up and call of SELF_TEST_post() function
HMAC-SHA2-256 (A3032)	SHA2-256 with a 256-bit key	KAT	CAST	Returns 1 on successful completion	Generate	Upon power-up and call of SELF_TEST_post() function
HMAC-SHA2-256 (A3252)	SHA2-256 with a 256-bit key	KAT	CAST	Returns 1 on successful completion	Generate	Upon power-up and call of SELF_TEST_post() function
KDF SP800-108 (A3032)	HMAC-SHA2-256	KAT	CAST	Returns 1 on successful completion	[S P800-108 rev1] Section 4.1 KAT for a Counter Mode KDF	Upon power-up and call of SELF_TEST_post() function
KDF SP800-108 (A3252)	HMAC-SHA2-256	KAT	CAST	Returns 1 on successful completion	[SP 800-108 rev1] Section 4.1 KAT for a Counter Mode KDF	Upon power-up and call of SELF_TEST_post() function
KDA OneStep SP800-56Cr2 (A3032)	SHA2-224	KAT	CAST	Returns 1 on successful completion	[SP 800-56C rev2] Section 4 OneStep KDF (AKA OpenSSL single-step or SS-KDF)	Upon power-up and call of SELF_TEST_post() function
KDA OneStep SP800-56Cr2 (A3252)	SHA2-224	KAT	CAST	Returns 1 on successful completion	[SP 800-56C rev2] Section 4 OneStep KDF (AKA OpenSSL single-step or SS-KDF)	Upon power-up and call of SELF_TEST_post() function

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDA TwoStep SP800-56Cr2 (A3032)	SHA2-256	KAT	CAST	Returns 1 on successful completion	[SP 800-56C rev2] Section 5 TwoStep KDF (HKDF variant)	Upon power-up and call of SELF_TEST_post() function
KDA TwoStep SP800-56Cr2 (A3252)	SHA2-256	KAT	CAST	Returns 1 on successful completion	[SP 800-56C rev2] Section 5 TwoStep KDF (HKDF variant)	Upon power-up and call of SELF_TEST_post() function
PBKDF (A3032)	SHA2-256, 24-byte password, 36-byte salt, iteration count of 4096	KAT	CAST	Returns 1 on successful completion	[SP 800-132] Section 5.3 KAT of Master Key derivation	Upon power-up and call of SELF_TEST_post() function
PBKDF (A3252)	SHA2-256, 24-byte password, 36-byte salt, iteration count of 4096	KAT	CAST	Returns 1 on successful completion	[SP 800-132] Section 5.3 KAT of Master Key derivation	Upon power-up and call of SELF_TEST_post() function
TLS v1.3 KDF (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[RFC8446] Section 7.1 TLS v1.3 KDF KAT	Upon power-up and call of SELF_TEST_post() function
TLS v1.3 KDF (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[RFC8446] Section 7.1 TLS v1.3 KDF KAT	Upon power-up and call of SELF_TEST_post() function
TLS v1.2 KDF RFC7627 (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 4.2.2 TLS 1.2 KAT	Upon power-up and call of SELF_TEST_post() function
TLS v1.2 KDF RFC7627 (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 4.2.2 TLS 1.2 KAT	Upon power-up and call of SELF_TEST_post() function

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
DSA KeyGen (FIPS186-4) (A3032)	PCT performed using the generated key pair	PCT	PCT	Returns 1 on successful completion	Sign, Verify	Performed on FFC (DSA, KAS-FFC-SSC) key pair generation, prior to returning the key pair on conclusion of the call
DSA KeyGen (FIPS186-4) (A3252)	PCT performed using the generated key pair	PCT	PCT	Returns 1 on successful completion	Sign, Verify	Performed on FFC (DSA, KAS-FFC-SSC) key pair generation, prior to returning the key pair on conclusion of the call
ECDSA KeyGen (FIPS186-4) (A3032)	PCT performed using the generated key pair	PCT	PCT	Returns 1 on successful completion	Sign, Verify	Performed on ECC (ECDSA, KAS-ECC CDH-Component, KAS-ECC-SSC) key pair generation, prior to returning the key pair on conclusion of the call
ECDSA KeyGen (FIPS186-4) (A3252)	PCT performed using the generated key pair	PCT	PCT	Returns 1 on successful completion	Sign, Verify	Performed on ECC (ECDSA, KAS-ECC CDH-Component, KAS-ECC-SSC) key pair generation, prior to returning the key pair on conclusion of the call
RSA KeyGen (FIPS186-4) (A3032)	PCT performed using the generated key pair	PCT	PCT	Returns 1 on successful completion	Sign, Verify	Performed on IFC (RSA, KAS-IFC-SSC, KTS-IFC) key pair generation, prior to returning the key pair on conclusion of the call
RSA KeyGen (FIPS186-	PCT performed using the	PCT	PCT	Returns 1 on successful	Sign, Verify	Performed on IFC (RSA, KAS-IFC-SSC, KTS-IFC)

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
4) (A3252)	generated key pair			completion		key pair generation, prior to returning the key pair on conclusion of the call
KDF ANS 9.42 (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.1 ANSI X9.42-2001 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF ANS 9.63 (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.1 X9.63-2001 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF IKEv2 (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 4.1.2 IKEv2 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SNMP (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.4 SNMPv3 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SRTP (A3032)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.3 SRTP KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SSH (A3032)	SHA1	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.2 SSHv2 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF ANS 9.42 (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.1 ANSI X9.42-2001 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF ANS 9.63 (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful	[SP 800-135 rev1] Section 5.1	Upon power-up and call of

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				completion	X9.63-2001 KDF KAT	SELF_TEST_post() function
KDF IKEv2 (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 4.1.2 IKEv2 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SNMP (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.4 SNMPv3 KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SRTP (A3252)	Fixed input KAT	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.3 SRTP KDF KAT	Upon power-up and call of SELF_TEST_post() function
KDF SSH (A3252)	SHA1	KAT	CAST	Returns 1 on successful completion	[SP 800-135 rev1] Section 5.2 SSHv2 KDF KAT	Upon power-up and call of SELF_TEST_post() function
TDES-CBC (A3032)	Keying Option: 1	KAT	CAST	Returns 1 on successful completion	Decrypt KAT	Upon power-up and call of SELF_TEST_post() function
TDES-CMAC (A3032)	Keying Option: 1	KAT	CAST	Returns 1 on successful completion	Verify KAT	Upon power-up and call of SELF_TEST_post() function
TDES-CBC (A3252)	Keying Option: 1	KAT	CAST	Returns 1 on successful completion	Decrypt KAT	Upon power-up and call of SELF_TEST_post() function
TDES-CMAC (A3252)	Keying Option: 1	KAT	CAST	Returns 1 on successful completion	Verify KAT	Upon power-up and call of SELF_TEST_post() function
AES-ECB Decrypt (A3032)	128 bits	KAT	CAST	Returns 1 on successful	Decrypt KAT	Upon power-up and call of

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				completion		SELF_TEST_post() function
AES-ECB Decrypt (A3252)	128 bits	KAT	CAST	Returns 1 on successful completion	Decrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-GCM Decrypt (A3032)	256 bits	KAT	CAST	Returns 1 on successful completion	Decrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-GCM Decrypt (A3252)	256 bits	KAT	CAST	Returns 1 on successful completion	Decrypt KAT	Upon power-up and call of SELF_TEST_post() function
AES-CMAC Verify (A3032)	128, 192, 256 bits	KAT	CAST	Returns 1 on successful completion	Verify KAT	Upon power-up and call of SELF_TEST_post() function
AES-CMAC Verify (A3252)	128, 192, 256 bits	KAT	CAST	Returns 1 on successful completion	Verify KAT	Upon power-up and call of SELF_TEST_post() function

Table 21: Conditional Self-Tests

The SELF_TEST_post() function performs all self-tests listed above with no operator intervention required when the module loads. The module returns a “1” if all self-tests succeed, and a “0” otherwise. The pre-operational and conditional self-tests may also be performed on-demand by calling this function and interpretation of the return code is the responsibility of the calling application.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3032)	Firmware Integrity Test	SW/FW Integrity	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A3252)	Firmware Integrity Test	SW/FW Integrity	On reboot or SELF_TEST_post() function call	Manual or reboot

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB Encrypt (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-ECB Encrypt (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-GCM Encrypt (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-GCM Encrypt (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-CMAC Generate (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-CMAC Generate (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
Counter DRBG (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
Counter DRBG (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
Hash DRBG (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
Hash DRBG (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
HMAC DRBG (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
HMAC DRBG (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
DSA SigGen (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
DSA SigGen (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
DSA SigVer (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
DSA SigVer (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA SigGen (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA SigGen (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA SigVer (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA SigVer (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA SigGen (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA SigGen (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA SigVer (FIPS186-4) (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA SigVer (FIPS186-4) (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KAS-FFC-SSC Sp800-56Ar3 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KAS-FFC-SSC Sp800-56Ar3 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KAS-ECC-SSC Sp800-56Ar3 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KAS-ECC-SSC Sp800-56Ar3 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KAS-IFC-SSC (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KAS-IFC-SSC (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA-1 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA-1 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA2-512 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA2-512 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA3-256 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
SHA3-256 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
HMAC-SHA2-256 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
HMAC-SHA2-256 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SP800-108 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SP800-108 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDA OneStep SP800-56Cr2 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDA OneStep SP800-56Cr2 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDA TwoStep SP800-56Cr2 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDA TwoStep SP800-56Cr2 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
PBKDF (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
PBKDF (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TLS v1.3 KDF (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TLS v1.3 KDF (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TLS v1.2 KDF RFC7627 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TLS v1.2 KDF RFC7627 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
DSA KeyGen (FIPS186-4) (A3032)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot
DSA KeyGen (FIPS186-4) (A3252)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA KeyGen (FIPS186-4) (A3032)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot
ECDSA KeyGen (FIPS186-4) (A3252)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA KeyGen (FIPS186-4) (A3032)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot
RSA KeyGen (FIPS186-4) (A3252)	PCT	PCT	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDF ANS 9.42 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF ANS 9.63 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF IKEv2 (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SNMP (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SRTP (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SSH (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF ANS 9.42 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF ANS 9.63 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF IKEv2 (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SNMP (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SRTP (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
KDF SSH (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TDES-CBC (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TDES-CMAC (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
TDES-CBC (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
TDES-CMAC (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-ECB Decrypt (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-ECB Decrypt (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-GCM Decrypt (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-GCM Decrypt (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-CMAC Verify (A3032)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot
AES-CMAC Verify (A3252)	KAT	CAST	On reboot or SELF_TEST_post() function call	Manual or reboot

Table 23: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	Error State is entered when self tests fail	Failure of self tests	Restarting the module	0

Table 24: Error States

If any self-test fails, an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the Approved mode if the module is reloaded and the call to SELF_TEST_post() succeeds. The CAST used to perform the approved integrity technique is passed before the execution of the pre-operational firmware integrity test (HMAC-SHA2-256).

10.5 Operator Initiation of Self-Tests

The operator can initiate the self-tests by calling SELF_TEST_post() or rebooting the host platform.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Per FIPS 140-3 classification, this is a multi-chip standalone cryptographic module. CiscoSSL FIPS Provider 8.0 is a C language-based firmware module that runs on production grade chassis. A complete revision history of the source code is collaborated by Bitbucket, and version controlled by Git. Code changes are tracked by commits tied to a username. All User documents are tracked in Cisco Document Central which requires username/password and access permission.
Coverity runs static analysis on the source code before committing to the secure repository.

Secure Distribution
The module is distributed only for use by Cisco personnel and as such is accessible only from the secure Cisco internal repository. Only authorized Cisco personnel have access to the module. The SHA512 fingerprint of the validated distribution tarball file can be obtained by contacting Cisco.

Secure Initialization
The module is ready to use after extracting it from the distribution tarball. The operating system loads the module into its user space. The initialization sequence starts with a check of the integrity of the runtime executable using a HMAC-SHA2-256 digest computed at build time. If the computed HMAC-SHA2-256 digest matches the stored known digest, then the cryptographic algorithm self-tests are performed. If any self-test fails, an internal global error flag is set to prevent subsequent invocation of any cryptographic function calls. Any such failure is a hard error that can only be recovered by reloading the module. Upon encountering a failure, the module will return an integer of 0. The module will only enter the Approved mode if the module is reloaded and the call to SELF_TEST_post() succeeds. The function call “. /openssl list -providers” returns the name and the version of the module.

Secure Operation
The tested operating systems segregate user processes into separate process spaces. Each process space is an independent virtual memory area that is logically separated from all other processes by the operating system firmware and hardware. The module functions entirely within the process space of the process that invokes it.

Additional information on switching between approved and non-approved mode is provided under “Mode Change Instructions and Status” in Section 2.4 of this SP.

11.2 Administrator Guidance

An additional guidance document, if required, can be obtained by contacting Cisco Systems, Inc. using the information posted on the validation certificate.

11.3 Non-Administrator Guidance

Not Applicable for this module.

11.4 Design and Rules

If CTR_DRBG is used, then the caller shall ensure that the derivation function is enabled.

12 Mitigation of Other Attacks

12.1 Attack List

The module implements two mitigations against timing-based side-channel attacks, namely Constant time Implementations and Blinding.

12.2 Mitigation Effectiveness

Constant-time Implementations protect cryptographic implementations in the Module against timing analysis since such attacks exploit differences in execution time depending on the cryptographic operation, and constant-time implementations ensure that the variations in execution time cannot be traced back to the key, CSP or secret data. Numeric Blinding protects the RSA, DSA and ECDSA algorithms from timing attacks. These algorithms are vulnerable to such attacks since attackers can measure the time of signature operations or RSA decryption.

To mitigate this the Module generates a random blinding factor which is provided as an input to the decryption/signature operation and is discarded once the operation has completed and resulted in an output. This makes it difficult for attackers to attempt timing attacks on such operations without the knowledge of the blinding factor and therefore the execution time cannot be correlated to the RSA/DSA/ECDSA key.