Arista Networks, Inc.

Arista Networks OpenSSL Module

Version 1.0.2h-fips

FIPS 140-2 Non-Proprietary Security Policy

Version 1.2

May 11, 2017

Copyright Notice

Sponsored by:

Intersoft International, Inc.

opengear
Advanced Console Management

PKWARE

PSW GROUP

CRYPTSOFT
sponsor of Beaglebone Black platforms

CITRIX®

Acknowledgments

The OpenSSL Software Foundation (OSF) serves as the "vendor" for this validation. Project management coordination for this effort was provided by:

Steve Marquess
The OpenSSL Software Foundation
1829 Mount Ephraim Road
Adamstown, MD 21710
USA
+1 877-673-6775
marquess@openssl.com

with technical work by:

Stephen Henson
4 Monaco Place,
Westlands, Newcastle-under-Lyme
Staffordshire. ST5 2QT.
England, United Kingdom
shenson@openssl.com
shenson@drh-consultancy.co.uk
http://www.drh-consultancy.co.uk/

Andy Polyakov
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
appro@openssl.org
appro@fy.chalmers.se

Tim Hudson
P.O. Box 6389
Fairfield Gardens 4103
Australia
ACN 074 537 821
tjh@openssl.com
tjh@cryptsoft.com
http://www.cryptsoft.com/

in coordination with the OpenSSL Team at www.openssl.org.

Validation testing was performed by InfoGard Laboratories. For information on validation or revalidations of software contact:

Marc Ireland
FIPS Program Manager, CISSP
InfoGard Laboratories
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401
805-783-0810 tel
805-783-0889 fax
mireland@infogard.com
http://www.infogard.com/

Modification History

2017-05-11	Updated per CMVP comments.
2017-01-31	Arista fork. Updated to comply with current standards (FIPS 186-4, SP800-131A, FIPS Annexes), tested against new OEs.
2015-06-29	Revised description for platforms 71, 72, 73, 74, 75, 76, 79, 80
2015-06-20	Revised description for platforms 66, 67
2015-06-16	Removed entries for platforms 47, 48, 49, 50, 59, 66, 67, 71, 72, 73, 74, 75, 76, 79, 80 to reflect prior CMVP action
2015-05-08	(2.0.10) Addition of nine platforms: #103/104 iOS 8.1 64-bit on Apple A7 (ARMv8) (without/with optimizations) #105 VxWorks 6.0 on Freescale P2020 (PPC) #106/107 iOS 8.1 32-bit on Apple A7 (ARMv8) (without/with optimizations) #108/109 Android 5.0 on Qualcomm APQ8084 (ARMv7) (without/with optimizations) #110/111 Android 5.0 64-bit on SAMSUNG Exynos7420 (ARMv8) (without/with optimizations)
2015-02-02	(2.0.9) Addition of new platform #102, TS-Linux 2.4 on ARMv4
2014-11-25	(2.0.9) Addition of new platforms #97, #98, VMware Horizon Workspace 2.1 x86 under vSphere Addition of new platform #99, QNX on ARMv4 Addition of new platforms #100, #101, Apple iOS 7.1 64-bit on ARMv8
2014-01-04	Addition of new platform #96, FreeBSD 8.4 on x86 without AES-NI
2014-07-30	Addition of two platforms #94, #95, FreeBSD 10.0 on x86, and re-removal of Dual EC DRBG
2014-07-28	Changed processor names for platforms #90, #91
2014-07-11	Added new platforms #88, #89, ArbOS 5.3 on x86 and #92, #93 FreeBSD 9.2 on x86
2014-06-12	Temporarily remove misplaced platform, move Dual EC DRBG to the Non-Approved Table 4c
2014-05-29	Added platforms #86, #87 FreeBSD 9.1 on x86, #90 Linux ORACLESP 2.6 on ASPEED AST-Series (ARMv5) , #91 ORACLESP 2.6 on Emulex PILOT 3 (ARMv5)
2014-05-12	Added platforms #81 Linux 2.6 on PPC, #82, #83 AcanOS 1.0 on x86, #84 AcanOS 1.0 on ARMv5, #85 FreeBSD 8.4 on x86 Multiple changes to separate the Approved services from those that are non Approved per the SP 800-131A transition
2013-11-08	Added two platforms #79, #80 PexOS 1.0 under vSphere with/without AES-NI
2013-11-01	Added two platforms #77, #78 iOS 6.0 with/without NEON
2013-10-02	Added six platforms (Linux 3.4 x86 virtualized under XenSource/VMware/Hyper-V, with/without AES-NI) Updated URL in Appendix A footnote
2013-08-29	Added new sponsor acknowledgment
2013-08-14	Added two Ubuntu 13.04 on ARMv7 (Beaglebone Black) and one Linux 3.8 on ARMv5TEJ platforms
2013-07-24	Added two VMware Horizon Workspace platforms

	Fixed typo in Table 4.1a, Hash DRBGs 888 bits not 880
2013-06-09	Added QNX, iOS 6.1, eCos for revision 2.0.5
2013-05-01	Added OpenWRT 2.6 for revision 2.0.4
2013-03-01	Added VMware Horizon Mobile 1.3, Apple OS X 10.7 , Apple iOS 5.0
2013-02-23	Added WinEC7 and Android 4.0 for revision 2.0.3
2013-02-14	Table 5: Removed references to non-existent Table 9 Table 4a: added certs Table 4.1a: Added AES GCM
2013-01-28	Added four platforms: Android 4.1 and Android 4.2 with and without NEON
2013-01-08	Reworded section 8
2013-01-03	Added Win2008, RHEL 32/64 bit under vSphere and Win7 with AES-NI.
2012-12-08	Note EC DH Key Agreement and RSA Key Wrapping strength.
2012-10-10	Added NetBSD 5.1 on PowerPC-e500, NetBSD 5.1 on Intel Xeon 5500 (x86-64) for revision 2.0.2
2011-07-02	Added DSP Media Framework, Linux 2.6/Freescale PowerPC-e500, Android 4.0
2011-06-15	Added iOS, WinCE 5, WinCE 6 OEs

References

Reference	Full Specification Name
[FIPS 140-2]	Security Requirements for Cryptographic modules, May 25, 2001
[FIPS 180-4]	Secure Hash Standard
[FIPS 186-4]	Digital Signature Standard
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[SP 800-38B]	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication
[SP 800-38C]	Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality
[SP 800-38D]	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC
[SP 800-56A]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[SP 800-67R1]	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-89]	Recommendation for Obtaining Assurances for Digital Signature Applications
[SP 800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[SP 800-131A]	Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

Table of Contents

1 Introduction

This document is the non-proprietary security policy for the Arista Networks OpenSSL Module, hereafter referred to as the Module.

The Module is a software library providing a C-language application program interface (API) for use by other processes that require cryptographic functionality. The Module is classified by FIPS 140-2 as a software module, multi-chip standalone module embodiment. The physical cryptographic boundary is the general purpose computer on which the module is installed. The logical cryptographic boundary of the Module is the OpenSSL API file. The Module performs no communications other than with the calling application (the process that invokes the Module services).

The FIPS 140-2 security levels for the Module are as follows:

Security Requirement	Security Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	NA
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	NA

Table 1 – Security Level of Security Requirements

The Module's software version for this validation is "openssl-1.0.2h-fips". This is a fork of the OpenSSL FIPS Object Module version 2.0.10.

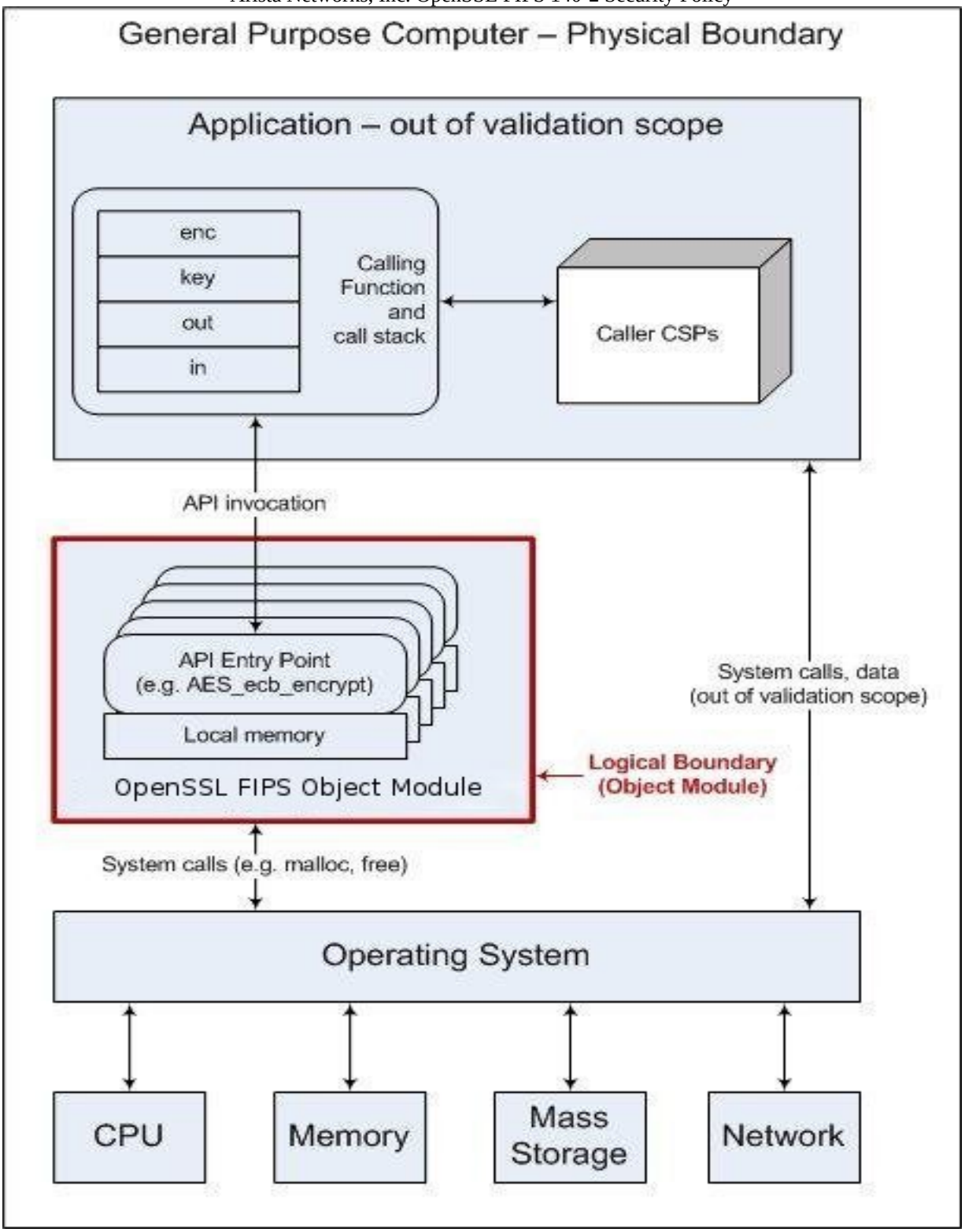

Figure 1 - Module Block Diagram

2 Module Configurations

The following configurations were operationally tested.

#	Operational Environment	Processor	Optimizations (Target)
1	EOS v4 on Arista 7150S	AMD Athlon NEO X2	None
2	EOS v4 on Arista 7508	Intel Sandy Bridge EN	None
3	EOS v4 on Arista 7308	Intel Broadwell-DE	None
4	EOS v4 on Arista 7010T	AMD G Series: eKabini	None
5	EOS v4 on Arista 7060CX	AMD G Series: Steppe Eagle	None

Table 2a - Tested Configurations

The following configurations have the same operating system and processors as the above configurations and are vendor affirmed to behave in the same manner. (As these have only been affirmed by the vendor, CMVP itself does not provide assurances to the correct operation of the module or the strength of generated keys.)

#	Operational Environment	Processor	Optimizations (Target)
1	EOS v4 on Arista 7050QX	AMD Athlon NEO X2	None
2	EOS v4 on Arista 7500R, 7320X, 7300X, 7250QX, 7280CR, 7050SX, 7050TX	Intel Sandy Bridge EN	None
3	EOS v4 on Arista 7504N, 7508N, 7512N	Intel Broadwell-DE	None
4	EOS v4 on Arista 7260, 7050SX, 7050TX, 7050QX	AMD G Series: eKabini	None
5	EOS v4 on Arista 7280QR, 7280SR, 7280TR, 7060, 7050SX, 7050TX	AMD G Series: Steppe Eagle	None

Table 2b – Vendor Affirmed Configurations

3 Ports and Interfaces

The physical ports of the Module are the same as the computer system on which it is executing. The logical interface is a C-language application program interface (API).

Logical interface type	Description
Control input	API entry point and corresponding stack parameters
Data input	API entry point data input stack parameters
Status output	API entry point return values and status stack parameters
Data output	API entry point data output stack parameters

Table 3 - Logical Interfaces

As a software module, control of the physical ports is outside module scope. However, when the module is performing self-tests, or is in an error state, all output on the logical data output interface is inhibited. The module is single-threaded and in error scenarios returns only an error value (no data output is returned).

4 Modes of Operation and Cryptographic Functionality

The Module supports only a FIPS 140-2 Approved mode and a non-Approved mode. The Approved mode is invoked by calling `FIPS_mode_set()` and using only Approved and allowed algorithms. Tables 4a and 4b list the Approved and non-approved but allowed algorithms, respectively.

Function	Algorithm	Options	Cert #
Random number generation	[SP 800-90] DRBG[1] Prediction resistance supported for all variations	Hash DRBG (SHA-1, all SHA-2 sizes) HMAC DRBG (SHA-1, all SHA-2 sizes) CTR DRBG (AES-128/192/256)	DRBG #1340
Encryption, Decryption and CMAC	[FIPS 197] AES [SP 800-38A] CBC, etc. [SP 800-38B] CMAC [SP 800-38C] CCM [SP 800-38D] GCM	128/192/256 ECB, CBC, OFB, CFB-1, CFB-8, CFB-128, CTR; CCM; GCM; CMAC	AES #4280
	[SP 800-67] Triple-DES	3-Key Triple-DES TECB, TCBC, TCFB-1, TCFB-8, TCFB-64, TOFB; CMAC	Triple-DES #2309
Message Digests	[FIPS 180-3] SHA-1/2	SHA-1, SHA-2 (224, 256, 384, 512)	SHS #3516
Keyed Hash	[FIPS 198] HMAC	SHA-1, SHA-2 (224, 256, 384, 512)	HMAC #2816
Digital Signature and Asymmetric Key Generation	[FIPS 186-4] RSA	GenKey9.31 (2048/3072) SigGen9.31, SigGenPKCS1.5, SigGenPSS (2048/3072 with all SHA2 sizes) SigVer9.31, SigVerPKCS1.5, SigVerPSS (2048/3072 with all SHA-2 sizes)	RSA #2301
	[FIPS 186-2] Legacy RSA	SigVer9.31, SigVerPKCS1.5, SigVerPSS (1024/1536/2048/3072/4096 with all SHA sizes)	RSA #2301
	[FIPS 186-4] DSA	PQG Gen, PQG Ver, Key Pair Gen, Sig Gen, Sig Ver (2048/3072) PQG Ver, Sig Ver (1024)	DSA #1141
	[FIPS 186-4] ECDSA	Functions: PKG, PKV, SigGen, SigGen Component, SigVer Curves: P-256, P-384, P-521 Hashes: SHA-1*, SHA-224, SHA-256, SHA-384, SHA-512 *SHA-1 signature generation is allowed for protocol usage only.	ECDSA #998
Key Derivation	[SP800-135] KDF, Application Specific[2]	TLS KDF (v1.0/v1.1 and v1.2) SSH KDF	CVL #1012

Table 4a – FIPS Approved Cryptographic Functions

Category	Algorithm	Description

[1] For all DRBGs the "supported security strengths" is just the highest supported security strength per [SP800-90A] and [SP800-57].

[2] Only the KDFs to these protocols have been tested by CAVP.

Key Agreement	EC Diffie-Hellman	Non-compliant (untested to SP800-56A) Diffie-Hellman scheme using elliptic curve, supporting curves P-256, P-384, and P-512. This key agreement scheme provides between 128 and 256 bits of encryption strength; however, it is only a service provided for calling process use. It is not used to establish keys into the Module.
Key Encryption, Decryption	RSA	The RSA algorithm may be used by the calling application for encryption or decryption of keys. It provides 112 or 128 bits of encryption strength. No claim is made for SP 800-56B compliance, and no CSPs are established into or exported out of the module using these services.
Hashing	MD5 within TLS	Component of TLS KDF

Table 4b – Non-FIPS Approved But Allowed Cryptographic Functions

The Module implements the following NIST-specified algorithms, which are non-Approved, either from algorithm transitions (e.g., SP800-131A) or from not being tested:

Function	Algorithm	Options
Encryption and Decryption	AES-XTS 128/256 (Untested)	Encrypt, decrypt
Digital Signature and Asymmetric Key Generation	RSA key size < 2048 (Disallowed)	GenKey9.31, SigGen9.31, SigGenPKCS1.5, SigGenPSS (<2048)
	DSA key size <2048 (Disallowed)	PQG Gen, Key Pair Gen, Sig Gen (<2048)
Key Encryption, Decryption	RSA key size < 2048 (Disallowed)	RSA key encryption/decryption (<2048)

Table 4c – Untested and Transition-Disallowed Cryptographic Functions

The algorithms in Table 4c must not be used when operating in the FIPS mode of operation.

The Module also implements the following algorithms, which are non-Approved:

Function	Algorithm	Function	Algorithm
Encryption and Decryption	AES/Triple-DES KW (non-compliant)	Encryption and Decryption	RC4
	Blowfish		RC5
	Camellia 128/192/256		SEED
	CAST5	Message Digests	MD4
	DES		MD5
	DES-X		RIPEMD-160
	IDEA		Whirlpool
	RC2	Keyed Hash	HMAC-MD5

Table 4d – Other non-Approved Cryptographic Functions

The algorithms in Table 4d are automatically disabled when in the FIPS mode of operation.

The Module is a cryptographic engine library, which can be used only in conjunction with additional software. Aside from the use of the NIST defined elliptic curves as trusted third party domain parameters, all other FIPS 186 assurances are outside the scope of the Module, and are the responsibility of the calling process.

4.1 Critical Security Parameters and Public Keys

All CSPs used by the Module are described in this section. All access to these CSPs by Module services are described in Section 4. The CSP names are generic, corresponding to API parameter data structures.

CSP Name	Description
AES EDK	AES encrypt / decrypt key
AES CMAC	AES CMAC generate / verify key
AES CCM	AES encrypt / decrypt / generate / verify key for CCM

AES GCM	AES encrypt / decrypt / generate / verify key for GCM
DRBG Seed	Entropy input for Hash, HMAC, or CTR DRBG.
Hash_DRBG CSPs	V (440/888 bits) and C (440/888 bits)
HMAC_DRBG CSPs	V (160/224/256/384/512 bits) and Key (160/224/256/384/512 bits)
CTR_DRBG CSPs	V (128 bits) and Key (AES 128/192/256)
DSA SGK	DSA signature generation key
ECDSA SGK	ECDSA signature generation key
EC Diffie-Hellman Private	EC Diffie-Hellman private key agreement key
HMAC Key	Keyed hash key (160/224/256/384/512)
RSA SGK	RSA signature generation key
RSA KDK	RSA key decryption (private key transport) key
Triple-DES EDK	Triple-DES (3-Key) encrypt / decrypt key
Triple-DES CMAC	Triple-DES (3-Key) CMAC generate / verify key

Table 4.1a – Critical Security Parameters

The module does not output intermediate key generation values.

Public Key Name	Description
DSA SVK	DSA signature verification key
ECDSA SVK	ECDSA signature verification key
EC Diffie-Hellman Public	EC Diffie-Hellman public key agreement key
RSA SVK	RSA signature verification public key
RSA KEK	RSA key encryption (public key transport) key

Table 4.1b – Public Keys

For all CSPs and Public Keys:

Storage: RAM, associated to entities by memory location. The Module stores RNG and DRBG state values for the lifetime of the RNG or 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), with the exception of RNG and DRBG state values used for the Module's default key generation service.

Generation: The Module implements SP 800-90A DRBG (Hash, HMAC, or CTR) services for creation of symmetric keys, and for generation of DSA, elliptic curve, and RSA keys as shown in Table 4a. The calling application is responsible for storage of generated keys returned by the module.

Entry: All CSPs enter the Module's logical boundary in plaintext as API parameters, associated by memory location. However, none cross the physical boundary.

Output: The Module does not output CSPs, other than as explicit results of key generation services. However, none cross the physical boundary.

Destruction: Zeroization of sensitive data is performed automatically by API function calls for temporarily stored CSPs. In addition, the module provides functions to explicitly destroy CSPs related to random number generation services. The calling application is responsible for parameters passed in and out of the module.

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 memory and process space from unauthorized access. Only the calling application that creates or imports keys can use or export such keys. All API functions are executed by the invoking calling application in a non-overlapping sequence such that no two API functions will execute concurrently. An

authorized application as user (Cryptographic Officer and User) has access to all key data generated during the operation of the Module.
In the event Module power is lost and restored the calling application must ensure that any AES GCM keys used for encryption or decryption are re-distributed.
For operation in the Approved mode, Module users (the calling applications) shall use entropy sources that contain at least 112 bits of entropy. To ensure full DRBG strength, the entropy sources must meet or exceed the security strengths shown in the table below.

DRBG Type	Underlying Algorithm	Minimum Seed Entropy
Hash_DRBG or HMAC_DRBG	SHA-1	128
	SHA-224	192
	SHA-256	256
	SHA-384	256
	SHA-512	256
CTR_DRBG	AES-128	128
	AES-192	192
	AES-256	256

Table 5 – DRBG Entropy Requirements

This entropy is supplied by means of callback functions. Those functions must return an error if the requested entropy strength cannot be met.

5 Roles, Authentication, and Services

The Module implements the required User and Cryptographic Officer roles and does not perform operator authentication.
Both roles have access to all of the services provided by the Module.

- User Role (User): Loading the Module and calling any of the API functions.
- Cryptographic Officer Role (CO): Installation of the Module on the host computer system and calling of any API functions.

All services implemented by the Module are listed below, along with a description of service CSP access. All services are available in both the Approved mode and the non-Approved mode. In the Approved mode, these services are restricted to the algorithms listed in Tables 4a and 4b. In the non-Approved mode, the algorithms listed in Tables 4c and 4d may also be used.

Service	Role	Description
Initialize	User, CO	Module initialization. Does not access CSPs.
Self-test	User, CO	Perform self-tests (FIPS_selftest). Does not access CSPs.
Show status	User, CO	Functions that provide module status information: ● Version (as unsigned long or const char *) ● FIPS Mode (Boolean) Does not access CSPs.
Zeroize	User, CO	Functions that destroy CSPs: ● fips_drbg_uninstantiate: for a given DRBG context, overwrites DRBG CSPs (Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs.) All other services automatically overwrite CSPs stored in allocated memory. Stack cleanup is the responsibility of the calling application.
Random number generation	User, CO	Used for random number and symmetric key generation. ● Seed or reseed a DRBG instance ● Determine security strength of a DRBG instance ● Obtain random data Uses and updates Hash_DRBG CSPs, HMAC_DRBG CSPs, CTR_DRBG CSPs.
Asymmetric key generation	User, CO	Used to generate DSA, ECDSA and RSA keys: RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK There is one supported security strength for each mechanism and algorithm type, the maximum specified in SP800-90
Symmetric encrypt/decrypt	User, CO	Used to encrypt or decrypt data. Executes using any symmetric encryption key from Table 4.1a: AES EDK, AES CCM, AES GCM, Triple-DES EDK (passed in by the calling process).
Symmetric digest	User, CO	Used to generate or verify data integrity with CMAC. Executes using AES CMAC, Triple-DES CMAC (passed in by the calling process).
Message digest	User, CO	Used to generate a SHA-1 or SHA-2 message digest. Does not access CSPs.
Keyed Hash	User, CO	Used to generate or verify data integrity with HMAC. Executes using HMAC Key (passed in by the calling process).

Key transport[3] (algorithms only)	User, CO	Used to encrypt or decrypt a key value on behalf of the calling process (the key is treated as payload data; this service does not establish keys into the module). Executes using RSA KDK, RSA KEK (passed in by the calling process).
Key agreement (algorithms only)	User, CO	Used to perform key agreement primitives on behalf of the calling process (this service does not establish keys into the module). Executes using EC Diffie-Hellman Private, EC Diffie-Hellman Public (passed in by the calling process).
Key derivation	User, CO	Used to perform key derivation primitives as per SP800-135: TLS KDF and SSH KDF (this service does not establish keys into the module).
Digital signature	User, CO	Used to generate or verify RSA, DSA or ECDSA digital signatures. Executes using RSA SGK, RSA SVK; DSA SGK, DSA SVK; ECDSA SGK, ECDSA SVK (passed in by the calling process).
Utility	User, CO	Miscellaneous helper functions. Does not access CSPs.

Table 6 - Services and CSP Access

[3] "Key transport" can refer to a) moving keys in and out of the module or b) the use of keys by an external application. The latter definition is the one that applies to the OpenSSL FIPS Object Module.

6 Self-Tests

The Module performs the self-tests listed below on invocation of Initialize or Self-Test.

Algorithm	Type	Test Attributes
Software integrity	KAT	HMAC-SHA1
HMAC	KAT	One KAT per SHA1, SHA224, SHA256, SHA384 and SHA512 Per IG 9.3, this testing covers SHA POST requirements.
AES	KAT	Separate encrypt and decrypt, ECB mode, 128 bit key length
AES CCM	KAT	Separate encrypt and decrypt, 192 key length
AES GCM	KAT	Separate encrypt and decrypt, 256 key length
AES CMAC	KAT	CMAC generate and verify, 128, 192, 256 key lengths
Triple-DES	KAT	Separate encrypt and decrypt, ECB mode, 3-Key
Triple-DES CMAC	KAT	CMAC generate and verify, 3-Key
RSA	KAT	Sign and verify using 2048 bit key, SHA-256, PKCS#1
DSA	PCT	Sign and verify using 2048 bit key, SHA-384
DRBG	KAT	CTR_DRBG: AES-256 with and without derivation function Hash_DRBG: SHA-256 HMAC_DRBG: SHA-256
ECDSA	PCT	Keygen, sign, verify using P-224 with SHA-512.

Table 7a - Power On Self Tests (KAT = Known answer test; PCT = Pairwise consistency test)

The Module is installed using one of the set of instructions in Appendix A, as appropriate for the target system. The HMAC-SHA-1 of the Module distribution file as tested by the CST Laboratory and listed in Appendix A is verified during installation of the Module file as described in Appendix A.

The `FIPS_mode_set()`[4] function performs all power-up self-tests listed above with no operator intervention required, returning a “1” if all power-up self-tests succeed, and a “0” otherwise. If any component of the power-up self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to `FIPS_mode_set()` succeeds.

The power-up self-tests may also be performed on-demand by calling `FIPS_selftest()`, which returns a “1” for success and “0” for failure. Interpretation of this return code is the responsibility of the calling application.

[4] `FIPS_mode_set()` calls Module function `FIPS_module_mode_set()`

The Module also implements the following conditional tests:

Algorithm	Test
DRBG	Health Tests as required by [SP800-90A] Section 11.3
DRBG	FIPS 140-2 continuous test (CRNGT) for stuck fault
DSA	Pairwise consistency test on each generation of a key pair
ECDSA	Pairwise consistency test on each generation of a key pair
RSA	Pairwise consistency test on each generation of a key pair

Table 7b - Conditional Tests

Pairwise consistency tests are performed for both possible modes of use, e.g. Sign/Verify and Encrypt/Decrypt. Failure of conditional self-tests is handled in the same manner as failure of power-on self-tests.

7 Operational Environment

The tested operating systems segregate user processes into separate process spaces. Each process space is logically separated from all other processes by the operating system software and hardware. The Module functions entirely within the process space of the calling application, and implicitly satisfies the FIPS 140-2 requirement for a single user mode of operation.

8 Mitigation of other Attacks

The module is not designed to mitigate against attacks which are outside of the scope of FIPS 140-2.