OpenSSL
Cryptography and SSL/TLS Toolkit

FIPS 140-3 Non-Proprietary Security Policy

OpenSSL FIPS Provider

Version**: 3.1.2**

Date: 29 December 2023

Copyright Notice

Modification History

Version	Description	Release Date
1.0	Initial Draft	29-Dec-2023

Table of Contents

List of Tables

List of Figures

1 General

i. FIPS 140-3 Overview

Federal Information Processing Standards Publication 140-3 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140-3 program. The National Voluntary Laboratory Accreditation Program (NVLAP) accredits independent testing labs to perform FIPS 140-3 testing; the CMVP validates modules meeting FIPS 140-3 security requirements. Validated is the term given to a module that is documented and tested against the FIPS 140-3 criteria.

More information is available on the CMVP website at: http://csrc.nist.gov/groups/STM/cmvp/index.html

ii. About this Document

This document describes the non-proprietary Security Policy for the OpenSSL FIPS Provider cryptographic module (hereafter referred to as "the Module") from The OpenSSL Project. It contains specification of the security rules under which the Module operates, including the security rules derived from the requirements of the FIPS 140-3 standard.

The OpenSSL Project may also be referred to as "OpenSSL" in this document.

The following trademarks are referenced within this Security Policy:

- Linux®: Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Unix®: UNIX is a registered trademark of The Open Group.
- Microsoft Windows®: Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

iii. Security Levels

The Module meets FIPS 140-3 overall Level 1 requirements, with security levels as follows:

Table 1 - Security Levels

ISO/IEC 24759 Section 6. [Number Below]	FIPS 140-3 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.	N/A
8	Non-invasive Security.	N/A
9	Sensitive Security Parameter Management.	1
10	Self-Tests.	1
11	Life Cycle Assurance.	3
12	Mitigation of Other Attacks.	1

In accordance with AS02.05, [ISO19790] §7.7 *Physical Security* is optional and does not apply to the Module.

In accordance with current CMVP policy, [ISO19790] §7.8 *Non-Invasive Security* is not applicable.

2 Cryptographic Module Specification

The Module is a cryptographic software library providing a C-language application program interface (API) for use by applications that require cryptographic functionality and is designated as a software module with a multi-chip standalone embodiment based on the descriptions of [ISO19790] AS02.03. The Module is intended for use by US and Canadian Federal agencies and other markets that require FIPS 140-3 validated cryptographic functionality.

The Module's formal name and version are "**OpenSSL FIPS Provider**" and "3.1.2", respectively.

The Module conforms to [FIPS140-3_IG] D.C *References to the Support of Industry Protocols*: while it provides [SP800-56Ar3] conformant schemes and API entry points oriented to TLS usage, the Module does not contain the full implementation of TLS. The following caveat is required:

> *No parts of the TLS protocol, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.*

The Module design corresponds to the Module security rules. Security rules enforced by the Module are described in the appropriate context of this document.

i. Cryptographic Boundary

Figure 1 depicts the Module operational environment, with the cryptographic boundary highlighted in red inclusive of all Module entry points (API calls). The Module is defined as a *Software module* per [ISO19790] AS02.03. The cryptographic boundary of the Module is the FIPS Provider, a dynamically loadable library. The Module performs no communication other than with the calling application via APIs that invoke the Module. The physical perimeter of the module is the General Purpose Computer (TOEPP).

No components are excluded from [FIPS140-3] requirements.

The pre-operational approved integrity test is performed over all components within the cryptographic boundary.

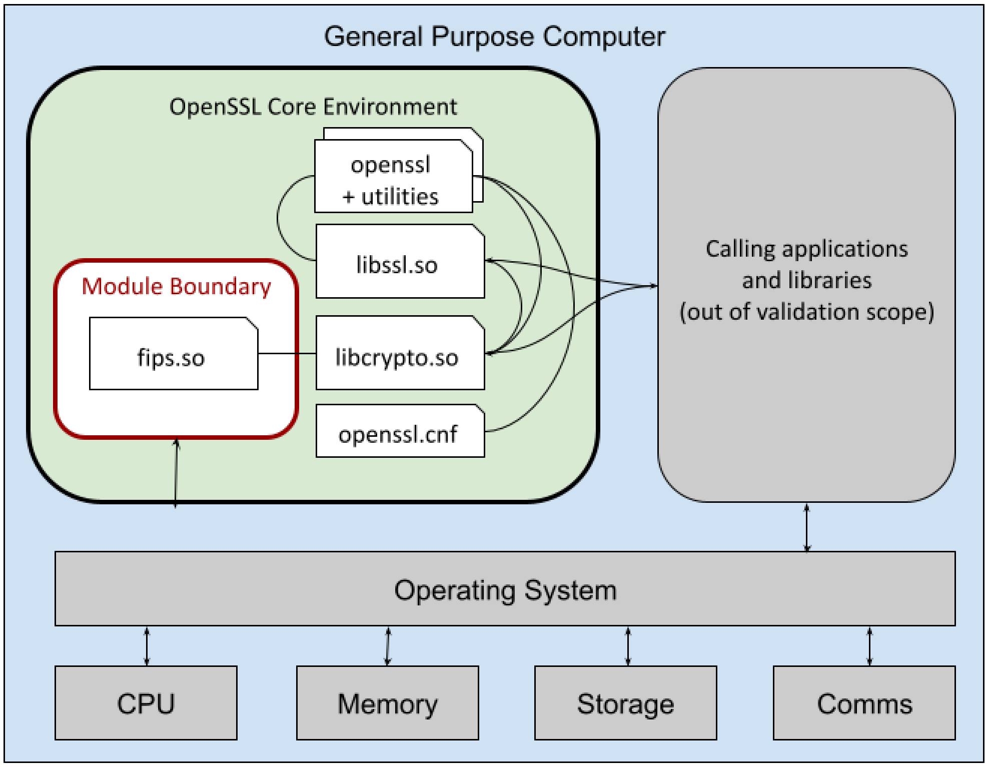

Figure 1 - Module Block Diagram

ii. Modes of Operation, Security Rules and Guidance

The Module supports an Approved mode and a non-Approved mode of operation. Use of the Approved algorithms listed in Table 3 and Non-Approved Algorithms Allowed in the Approved Mode listed in Table 4 will place the module in the Approved mode of operation. Use of the non-Approved Algorithms Not Allowed in the Approved Mode listed in Table 5 will place the module in the non-Approved mode of operation.

The inherent properties of the Module are:

1. Manual key entry is not supported.
2. Data output is inhibited during self-tests, zeroisation, SSP generation and error states.
3. The Module does not perform any cryptographic function if any self-test has failed.

The conditions for using the Module in the [FIPS140-3] Approved mode of operation are:

1. Installation of the Module as described in Section 11 results in the settings described below, which are required for operation in the Approved mode:

    a. security-checks = 1

    Enforce minimum key strengths and approved curve names.

    b. conditional-errors = 1

    Enforce the Module entering the error state on conditional test errors such as PCT failure.

    c. drbg-no-trunc-md=1

    Disallow use of truncated digests with HASH and HMAC DRBGs (IG D.R)

    d. tls1-prf-ems-check=1

    Enforce Extended Master Secret (EMS) use with TLS 1.2 (IG D.Q)

2. The Module is a cryptographic library used by a calling application. The calling application is responsible for:
    a. Use of the primitives in the correct sequence.
    b. Use of keys in accordance with [SP800-140Dr2] (as the keys used by the Module for cryptographic purposes are provided over the call stack by the calling application).
    c. Use of a [SP800-90B] compliant entropy source with at least 256 bits of security strength. Entropy is supplied to the Module via callback functions. The callback functions shall return an error if the minimum entropy strength cannot be met.

The Module obtains the [FIPS140-3_IG] D.F required key agreement assurances:

- [SP800-56Ar3] in accordance with Section 5.6.2.
- [SP800-56Br2] in accordance with Section 6.4.

a. AES-GCM Usage

The Module supports internal AES GCM IV generation compliant to [FIPS140-3_IG] C.H *Key/IV Pair Uniqueness Requirements from SP 800-38D* Scenario 1(a), tested per option (ii) under C.H TLS/DTLS 1.2 protocol IV generation**,** Scenario 1(d) SSHv2 per RFC4252, RFC4253 and RFC5647 and Scenario 5 TLS 1.3 per RFC8446.

The Module does not implement the TLS and SSH protocols itself, however, it provides the cryptographic functions required for implementing the protocols. AES GCM encryption is used in the context of the TLS protocol versions 1.2 and 1.3.For TLS v1.2, the mechanism for IV generation is compliant with RFC 5288. The module provides the primitives to support the AES GCM ciphersuites from [SP800-52r1] Section 3.3.1. The counter portion of the IV is strictly increasing. When the IV exhausts the maximum number of possible values for a given session key, this results in a failure in encryption and a handshake to establish a new encryption key will be required. It is the responsibility of the user of the module, i.e., the first party, client or server, to encounter this condition, to trigger this handshake in accordance with RFC 5246. For TLS v1.3, the mechanism for IV generation is compliant with RFC 8446.

The Module also supports internal IV generation using the module's approved DRBG. The IV is at least 96 bits in length per [SP800-38D] Section 8.2.2. Per [FIPS140-3_IG] C.H Scenario 2 and [SP800-38D], the approved DRBG generates outputs such that the (key, IV) pair collision probability is less than 2^{-32}.

In each case, in the event that the Module power is lost and restored the user must ensure that the AES GCM encryption/decryption keys are re-distributed. The module does not support persistent storage of SSPs.

The Module also supports importing of GCM IVs when an IV is not generated within the Module. In the approved mode, an IV must not be imported for encryption from outside the cryptographic boundary of the Module as this will result in a non-conformance.

b. PBKDF Usage

The supported lengths of a password/passphrase used in key derivation can range between 8-128 bits. The iteration count values used range from 1 to 10000 per NIST SP 800-132 Section 5.2 whereby the iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. Keys derived from passwords, as shown in SP 800-132, may only be used in storage applications. The security strength of the derived key is at least 112 bits.

c. AES-XTS Usage

In accordance with [SP800-38E], the XTS-AES algorithm shall only be used for confidentiality on storage devices. The Module complies with [FIPS140-3_IG] C.I by explicitly checking that Key_1 ≠ Key_2 before using the keys in the XTS-AES algorithm to process data with them.

iii. Degraded Operation

The Module does not implement a degraded mode.

iv. Operational Environment

Operational testing was performed for the following Operating Environments:

Table 2 - Tested Operational Environments

#	Operating System	Hardware Platform	Processor	PAA/Acceleration
1	Ubuntu Linux 22.04.1 Server.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	Without PAA.
2	Ubuntu Linux 22.04.1 Server.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	With PAA.
3	Debian 11.5.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	Without PAA.
4	Debian 11.5.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	With PAA.
5	FreeBSD 13.1.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	Without PAA.
6	FreeBSD 13.1.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	With PAA.
7	Windows 10.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	Without PAA.
8	Windows 10.	Dell Inspiron 7591 2 in 1.	Intel i7-8550U.	With PAA.
9	macOS 11.5.2.	Apple M1 Mac Mini.	M1.	Without PAA.
10	macOS 11.5.2.	Apple M1 Mac Mini.	M1.	With PAA.
11	macOS 11.5.2.	Apple i7 Mac Mini.	Intel i7.	Without PAA.
12	macOS 11.5.2.	Apple i7 Mac Mini.	Intel i7.	With PAA.

The Module conforms to [FIPS140-3_IG] 2.3.C *Processor Algorithm Accelerators (PAA) and Processor Algorithm Implementation (PAI).* The AES-NI functions are identified by [FIPS140-3_IG] 2.3.C as a known PAA.

No operational environments are vendor affirmed.

v. Approved and Allowed Cryptographic Functionality

The Module implements the Approved cryptographic functions listed in Table 3, organized for consistency with the categorization of operations inherent to the Module's *fips_query* function. Equivalent strength in bits is given for each key or algorithm type (as some algorithms do not use or produce keys). The term *s* is used throughout to indicate security strength, following the notation used in the majority of the sources (refer to the notes below Table 3). This table is referenced by Table 10 (SSPs).

Table 3 - Approved Algorithms

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A3548	AES [FIPS197], [SP800-38A]	AES-CBC, AES-CFB1, AES-CFB128, AES-CFB8, AES-CTR, AES-ECB, AES-OFB.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Encryption, Decryption.
A3548	AES [FIPS197], [SP800-38A_Add]	AES-CBC-CS1, AES-CBC-CS2, AES-CBC-CS3.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Encryption, Decryption.
A3548	AES [SP800-38C]	AES-CCM.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Authenticated Encryption, Authenticated Decryption, Message Authentication.
A3548	AES [SP800-38B]	AES-CMAC.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Generation, Verification.
A3548	AES [SP800-38D]	AES-GCM, AES-GMAC.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Authenticated Encryption, Authenticated Decryption, Message Authentication.
A3548	AES [SP800-38F]	AES-KW, AES-KWP. Cipher, Inverse.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Key Wrap, Key Wrap with Padding.
A3548	AES [SP800-38E]	AES-XTS Testing Revision 2.0.	AES-128 (s = 128), AES-256 (s = 256).	Encryption, Decryption.
Vendor Affirmed	CKG [SP800-133r2]	§4: Using the Output of a Random Bit Generator. §5.1 Key Pairs for Digital Signature Schemes. §5.2 Key Pairs for Key Establishment. §6.1: Direct Generation of Symmetric Keys. §6.2: Derivation of Symmetric Keys.	N/A.	Cryptographic Key Generation.
A3548	Counter DRBG [SP800-90Ar1]	Tested with derivation function enabled and disabled; prediction resistance supported.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256).	Random Number Generation, Symmetric Key Generation.

A3548	DSA KeyGen [FIPS186-4]	FFC Key Generation.	L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				Key Generation.

A3548	DSA PQGGen [FIPS186-4]	P/Q Generation: Probable.
		G Generation: Canonical, Unverifiable.
		Tested with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.
			L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				PQG Generation.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function

Vendor Affirmed	DSA PQGGen [FIPS186-4]	PQGGen using SHA3; no ACVP testing is available.	L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				PQG Generation.

A3548	DSA PQGVer [FIPS186-4]	P/Q Generation: Probable.
		G Generation: Canonical, Unverifiable.
		Tested with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.
			L = 1024/N = 160 (s < 112)
			L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				PQG Verification. (Verification with SHA-1 and L=1024/N=160 is for legacy use only.)

Vendor Affirmed	DSA PQGVer [FIPS186-4]	PQGVer using SHA3; no ACVP testing is available.	L = 1024/N = 160 (s < 112)
			L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				PQG Verification. (Verification with L=1024/N=160 is for legacy use only.)

A3548	DSA SigGen [FIPS186-4]	SigGen (tested with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256).	L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				Signature Generation.

Vendor Affirmed	DSA SigGen [FIPS186-4]	SigGen using SHA3; no ACVP testing is available.	L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				Signature Generation.

A3548	DSA SigVer [FIPS186-4]	SigVer (tested with SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256).	L = 1024/N = 160 (s < 112)
			L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				Signature Verification. (Verification with SHA-1 and L=1024/N=160 is for legacy use only.)

Vendor Affirmed	DSA SigVer [FIPS186-4]	SigVer using SHA3; no ACVP testing is available.	L = 1024/N = 160 (s < 112)
			L = 2048/N = 224 (s = 112),
			L = 2048/N = 256 (s = 112)
			L = 3072/N = 256 (s = 128).
			See Note 5 and Note 9.
				Signature Verification. (Verification with L=1024/N=160 is for legacy use only.)

A3548	ECDSA KeyGen [FIPS186-4]	Secret Generation Mode: Testing Candidates.	B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2 and Note 3.
				ECC Key Generation.

A3548	ECDSA KeyVer [FIPS186-4]	Public Key Validity.	B-163, K-163, P-192 (s < 112);
			B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2 and Note 3.
				ECC Public Key Validation. (Curves B-163, K-163, and P-192 are for legacy use only.)

CVL
A3548
	ECDSA SigGen [FIPS186-4]	SigGen Component (tested with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512).	B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2 and Note 3.
				Signature Generation Component.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function

A3548	ECDSA SigGen
	[FIPS186-4]
		SigGen (tested with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512).	B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2 and Note 3.
				Signature Generation.

A3548	ECDSA SigVer
	[FIPS186-4]
		SigVer (tested with SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512).	B-163, K-163, P-192 (s < 112);
			B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2 and Note 3.
				Signature Verification. (Verification with SHA-1 and curves B-163, K-163, and P-192 is for legacy use only.)

A3548	Hash DRBG [SP800-90Ar1]	Tested with prediction resistance enabled.	SHA-1 (s = 160), SHA2-256 (s = 256), SHA2-512 (s = 512).	Random Number Generation, Symmetric Key Generation.
A3548	Hash DRBG [SP800-90Ar1]	Hash DRBG using SHA3 modes.	SHA3-256 (s = 256), SHA3-512 (s = 512).	Random Number Generation, Symmetric Key Generation.
A3548	HMAC DRBG [SP800-90Ar1]	Tested with prediction resistance enabled.	SHA-1 (s = 160), SHA2-256 (s = 256), SHA2-512 (s = 512).	Random Number Generation, Symmetric Key Generation.
A3548	HMAC DRBG [SP800-90Ar1]	HMAC DRBG using SHA3 modes.	SHA3-256 (s = 256), SHA3-512 (s = 512).	Random Number Generation, Symmetric Key Generation.
A3548	HMAC-SHA-1 [FIPS198-1]	Generate HMAC-SHA-1 MAC with SHA-1.	SHA-1 (s = 160).	Generation, Verification, Message Authentication.
A3548	HMAC-SHA2 [FIPS198-1]	Generate HMAC-SHA2 MAC with the listed SHA2 modes.	SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256).	Generation, Verification, Message Authentication.
A3548	HMAC-SHA3 [FIPS198-1]	Generate HMAC-SHA3 MAC with the listed SHA3 modes.	SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512).	Generation, Verification, Message Authentication.

CVL
A3548
	KAS-ECC CDH-Component
	[SP800-56Ar3]
		KAS-ECC CDH Component with the listed ECC key types.	B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2, Note 3 and Note 4.
				§5.7.1.2 ECC CDH Primitive used in Shared Secret Computation.

A3548	KAS-1
	KAS-ECC-SSC
	[SP800-56Ar3]
	IG D.F Scenario 2 path (1)
		Scheme: ephemeralUnified.
		KAS Role: Initiator, Responder.
			B-233, K-233, P-224 (s ~= 112);
			B-283, K-283, P-256 (s ~= 128);
			B-409, K-409, P-384 (s ~= 192);
			B-571, K-571, P-521 (s ~= 256).
			See Note 2, Note 3 and Note 4.
			SSP establishment methodology provides between 112 and 256 bits of encryption strength).
				Key Agreement primitives.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A3548	KAS-2 KAS-FFC-SSC [SP800-56Ar3] IG D.F Scenario 2 path (1)	Scheme: dhEphem. KAS Role: Initiator, Responder.	FB (s = 112), FC (s = 112), ffdhe2048 (s = 112), ffdhe3072 (112 ≤ s ≤ 128), ffdhe4096 (112 ≤ s ≤ 152), ffdhe6144 (112 ≤ s ≤ 176), ffdhe8192 (112 ≤ s ≤ 200), MODP-2048 (s = 112), MODP-3072 (112 ≤ s ≤ 128), MODP-4096 (112 ≤ s ≤ 152), MODP-6144 (112 ≤ s ≤ 176), MODP-8192 (112 ≤ s ≤ 200). See Note 6. SSP establishment methodology provides between 112 and 200 bits of encryption strength).	Key Agreement primitives.

A3548	KAS-3
	KAS-IFC-SSC
	[SP800-56Br2]
	IG D.F Scenario 1 path (1)
		Scheme: KAS1, KAS2.
		KAS Role: Initiator, Responder.
			k=2048 (s ~= 112), k=3072 (s ~= 128),
			k=4096 (s ~= 152), k=6144 (s ~= 176),
			k=8192 (s ~= 200).
			See Note 7.
			
			SSP establishment methodology provides between 112 and 200 bits of encryption strength).
				Key Agreement primitives.

A3548	KDA HKDF [SP800-56Cr2]	HMAC algorithm with the listed SHA variants.	SHA-1 (s = 160), SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256), SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512).	Key Derivation.
A3548	KDA OneStep [SP800-56Cr2]	One-Step KDF with the listed auxiliary functions.	SHA-1 (s = 160), SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256), SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512); HMAC-SHA-1 (s = 160), HMAC-SHA2-224 (s = 224), HMAC-SHA2-256 (s = 256), HMAC-SHA2-384 (s = 384), HMAC-SHA2-512 (s = 512), HMAC-SHA2-512/224 (s = 224), HMAC-SHA2-512/256 (s = 256), HMAC-SHA3-224 (s = 224), HMAC-SHA3-256 (s = 256), HMAC-SHA3-384 (s = 384), HMAC-SHA3-512 (s = 512); KMAC-128 (112 ≤ s ≤ 128), KMAC-256 (112 ≤ s ≤ 256). See Note 8.	Key Derivation.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A3548	KDA TwoStep [SP800-56Cr2]	Two-Step KDF. KDF Mode: Feedback mode with the listed MAC modes.	HMAC-SHA-1 (s = 160), HMAC-SHA2-224 (s = 224), HMAC-SHA2-256 (s = 256), HMAC-SHA2-384 (s = 384), HMAC-SHA2-512 (s = 512), HMAC-SHA2-512/224 (s = 224), HMAC-SHA2-512/256 (s = 256), HMAC-SHA3-224 (s = 224), HMAC-SHA3-256 (s = 256), HMAC-SHA3-384 (s = 384), HMAC-SHA3-512 (s = 512).	Key Derivation.
CVL A3548	KDF ANS 9.42 [SP800-135r1]	KDF ANS 9.42 with the listed hash algorithms.	SHA-1 (s = 160), SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256), SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512).	Key Derivation.
CVL A3548	KDF ANS 9.63 [SP800-135r1]	KDF ANS 9.63 with the listed hash algorithms.	SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512).	Key Derivation.
A3548	KDF KMAC [SP800-108r1]	KDF with the listed KMAC modes.	KMAC-128 (112 ≤ s ≤ 128), KMAC-256 (112 ≤ s ≤ 256).	Key Derivation.
A3548	KDF [SP800-108r1]	KDF modes: Counter and Feedback, with the listed MAC modes.	CMAC-AES128 (s = 128), CMAC-AES192 (s = 192), CMAC-AES256 (s = 256), HMAC-SHA-1 (s = 160), HMAC-SHA2-224 (s = 224), HMAC-SHA2-256 (s = 256), HMAC-SHA2-384 (s = 384), HMAC-SHA2-512 (s = 512), HMAC-SHA2-512/224 (s = 224), HMAC-SHA2-512/256 (s = 256), HMAC-SHA3-224 (s = 224), HMAC-SHA3-256 (s = 256), HMAC-SHA3-384 (s = 384), HMAC-SHA3-512 (s = 512).	Key-Based Key Derivation.
CVL A3548	KDF SSH [SP800-135r1]	Derivation of key blocks for the listed AES cipher key types and hash algorithms.	AES-128 (s = 128), AES-192 (s = 192), AES-256 (s = 256); SHA-1 (s = 160), SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512).	Key Derivation.
A3548	KMAC [SP800-185]	KMAC, with optional XOF support, without hex customization string.	KMAC-128 (112 ≤ s ≤ 128), KMAC-256 (112 ≤ s ≤ 256). See Note 8.	Message Authentication.
A3548	KTS-1 [SP800-38F]	AES CCM; AES GCM; AES KW, KWP.	Key Transport (AES Cert. #A3548; SSP establishment methodology provides between 128 and 256 bits of encryption strength).	Key Transport in compliance with [SP800-38F] when approved using an Authenticated AES mode.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A3548	KTS-2 [SP800-38F]	AES (any mode) and HMAC.	Key Transport (AES Cert. #A3548 and HMAC Cert. #A3548; SSP establishment methodology provides between 128 and 256 bits of encryption strength).	Key Transport in compliance with [SP800-38F] when approved AES and approved HMAC are used in combination.
A3548	KTS-3 [SP800-38F]	AES (any mode) and CMAC, GMAC.	Key Transport (AES Cert. #A3548 and AES Cert. #A3548; SSP establishment methodology provides between 128 and 256 bits of encryption strength).	Key Transport in compliance with [SP800-38F] when approved AES and approved CMAC/GMAC are used in combination.
A3548	KTS-4 KTS-IFC [SP800-56Br2]	Scheme: KTS-OAEP-basic (no key confirmation): RSA-OAEP, RSADP, RSAEP, Key Encapsulation, Key Unencapsulation Key Generation Methods: rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor.	k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152), k=6144 (s ~= 176). SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512. See Note 5 and Note 7.	Key Transport.
A3548	PBKDF [SP800-132]	Option 1a, per Section 5.4 using HMAC and the SHAs listed at right.	SHA-1 (s = 160), SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256), SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512).	Password-Based Key Derivation.
A3548	RSA KeyGen [FIPS186-4]	Key generation mode: B.3.6. Primality tests per Tables C.2 and C.3, with listed moduli.	k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Key Generation.
A3548	RSA SigGen [FIPS186-4]	Signature type: ANSI X9.31 tested with the listed moduli and the following hash algorithms: SHA2-256, SHA2-384, SHA2-512.	k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Generation.
A3548	RSA SigGen [FIPS186-4]	Signature type: PKCS 1.5 tested with the listed moduli and the following hash algorithms: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.	k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Generation.
A3548	RSA SigGen [FIPS186-4]	Signature type: PKCSPSS tested with the listed moduli and the following hash algorithms: SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.	k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Generation.
CVL A3548	RSA Signature Primitive [FIPS186-4]	Private Key format: CRT. Public Exponent Mode: Fixed.	k = 2048. See Note 5 and Note 7.	Signature primitive.
A3548	RSA SigVer [FIPS186-4]	Signature type: ANSI X9.31 tested with the listed moduli and the following hash algorithms: SHA-1*, SHA2-256, SHA2-384, SHA2-512.	k=1024 (s ≤ 112), k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Verification. (Verification with SHA-1 and modulus length k=1024 is for legacy use only.)

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
A3548	RSA SigVer [FIPS186-4]	Signature type: PKCS 1.5 tested with the listed moduli and the following hash algorithms: SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.	k=1024 (s ≤ 112), k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Verification. (Verification with SHA-1 and modulus length k=1024 is for legacy use only.)
A3548	RSA SigVer [FIPS186-4]	Signature type: PKCSPSS tested with the listed moduli and the following hash algorithms: SHA-1*, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256.	k=1024 (s ≤ 112), k=2048 (s ~= 112), k=3072 (s ~= 128), k=4096 (s ~= 152). See Note 5 and Note 7.	Signature Verification. (Verification with SHA-1 and modulus length k=1024 is for legacy use only.)
A3548	Safe Primes Key Generation [SP800-56Ar3], [RFC7919]	Safe Primes (FFC) key generation for the listed groups.	ffdhe2048 (s = 112), ffdhe3072 (112 ≤ s ≤ 128), ffdhe4096 (112 ≤ s ≤ 152), ffdhe6144 (112 ≤ s ≤ 176), ffdhe8192 (112 ≤ s ≤ 200), MODP-2048 (s = 112), MODP-3072 (112 ≤ s ≤ 128), MODP-4096 (112 ≤ s ≤ 152), MODP-6144 (112 ≤ s ≤ 176), MODP-8192 (112 ≤ s ≤ 200). See Note 6.	Key Generation.
A3548	Safe Primes Key Verification [SP800-56Ar3], [RFC7919]	Safe Primes (FFC) key verification for the listed groups.	ffdhe2048 (s = 112), ffdhe3072 (112 ≤ s ≤ 128), ffdhe4096 (112 ≤ s ≤ 152), ffdhe6144 (112 ≤ s ≤ 176), ffdhe8192 (112 ≤ s ≤ 200), MODP-2048 (s = 112), MODP-3072 (112 ≤ s ≤ 128), MODP-4096 (112 ≤ s ≤ 152), MODP-6144 (112 ≤ s ≤ 176), MODP-8192 (112 ≤ s ≤ 200). See Note 6.	Key Verification.
A3548	SHA-1 [FIPS180-4]	SHA-1 mode listed at right.	SHA-1 (s = 160). See Note 1. Large Message Sizes: 1, 2, 4, 8gigabytes	Message Digest Generation.
A3548	SHA2 [FIPS180-4]	SHA2 modes listed at right.	SHA2-224 (s = 224), SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512), SHA2-512/224 (s = 224), SHA2-512/256 (s = 256). See Note 1. Large Message Sizes: 1, 2, 4, 8gigabytes	Message Digest Generation.
A3548	SHA3 [FIPS202]	SHA3 modes listed at right.	SHA3-224 (s = 224), SHA3-256 (s = 256), SHA3-384 (s = 384), SHA3-512 (s = 512). See Note 1. Large Message Sizes: 1, 2, 4, 8gigabytes	Message Digest Generation.
A3548	SHAKE [FIPS202]	SHAKE extendable-output function modes listed at right.	SHAKE-128 (s = 128), SHAKE-256 (s = 256). See Note 1.	Message Digest Generation.
CVL A3548	TLS v1.2 KDF RFC7627	TLS [RFC7627] key derivation with Extended Master Secret (EMS) support, using the listed hash algorithms.	SHA2-256 (s = 256), SHA2-384 (s = 384), SHA2-512 (s = 512).	Key Derivation for use with the TLS v1.2 protocol.

CAVP Cert	Algorithm and Standard	Mode/Method	Description / Key Size(s) / Key Strength(s)	Use / Function
CVL A3548	TLS v1.3 KDF [RFC8446]	KDF running modes: DHE, PSK, PSK-DHE, using the listed HMAC algorithms.	HMAC-SHA2-256 (s = 256), HMAC-SHA2-384 (s = 384).	Key Derivation for use with the TLS v1.3 protocol.

* CAVP testing permits testing with SHA-1.

Note 1: Preimage resistance strength applies to hash algorithms used in DRBG, KDFs. Described also in [SP800-57P1r5] Table 3.

Note 2: Elliptic curve strengths are annotated as approximate (i.e., s ~=) since [SP800-186] Table 1 provides approximate security strengths.

Note 3: [SP800-186] (cited in [SP800-140Cr2]) and [FIPS140-3_IG] C.K indicate that the Binary (B-) and Koblitz (K-) curves are deprecated.

Note 4: Approved elliptic curves for ECC key agreement are given in [SP800-56Ar3] Table 24.

Note 5: In Digital Signature applications, security strength is primarily associated with the asymmetric key pair specification. The hash function used must have equivalent strength equal to or greater than the security strength of the associated key pair.

Note 6: Approved key types for FFC key agreement are given in [SP800-56Ar3] Tables 25, 26. The group notation of Table 26 is used for consistency with CAVP algorithm listings and ACVP capability registration.

Note 7: Approved key types for IFC key agreement are given in [SP800-56Br2] Table 4. IFC key types approved for Digital Signature Generation and Verification are given also in [SP800-57P1r5] Table 2. Equivalent strengths are annotated as approximate (i.e., s ~=) since [SP800-56Br2] Table 4 provides approximate security strengths.

Note 8: Security strengths for KDA One Step are given in [SP800-56Cr2] Table 1 (hash), Table 2 (HMAC) and Table 3 (KMAC).

Note 9: Security strength for L=2048/N=256 is determined in accordance with [FIPS140-3_IG] D.B *Strength of SSP Establishment Methods* as y = min(x, N/2), where x is 112 and therefore y = min(112, 128) = 112.

Reference sources for the strengths provided in Table 3 are as follows:

- AES (AES-128, AES-192, AES-256): [SP800-57P1r5] Table 2.
- ECC (B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521): [SP800-186] Table 1.
- FFC (L=1024/N=160, L=2048/N=224, L=2048/N=256, L=3072/N=256): [SP800-57P1r5] Table 2.
- FFC (ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192): [SP800-56Ar3] Tables 25 and 26.
- IFC (k=1024, k=2048, k=3072, k=4096, k=6144, k=8192): [SP800-56Br2] Table 4.
- KMAC (KMAC128, KMAC256): [SP800-56Cr2] Table 3.
- SHA-1, SHA2 (SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256): [SP800-107] Table 1.
- SHA3 (SHA3-224, SHA3-256, SHA3-384, SHA3-512): [SP800-57P1r5] Table 3.
- SHAKE (SHAKE128, SHAKE256): [SP800-185] Section 8.1.

Table 4 - Non-Approved Algorithms Allowed in the Approved Mode of Operation

Algorithm	Caveat	Use/Function
AES	Cert. A3548, key unwrapping. Per IG D.G.	Symmetric key unwrapping.

Table 5 - Non-Approved Algorithms Not Allowed in the Approved Mode of Operation

Algorithm/Function	Use/Function
Triple-DES	Provides 3-Key ECB and CBC mode, but indicated as *fips=no*, Encryption, Decryption.
Ed448	SHAKE256, Ed448 provides 224 bits of security, Digital Signature Generation.
Ed25519	SHA2-512, Ed25519 provides 128 bits of security, Digital Signature Generation.
X448	Provides 224 bits of security, Key Agreement.

X25519	Provides 128 bits of security, Key Agreement.

3 Cryptographic Module Interfaces

Table 6 defines the Module's [FIPS140-3] logical interfaces; the Module does not interact with physical ports.

Table 6 - Ports and Interfaces

Logical Interface	Data that Passes over Port/Interface
Control input.	API entry point: stack frame including non-sensitive parameters.
Data input.	API call parameters passed by reference or value for cryptographic service input.
Status output.	API return value: enumerated status resulting from call execution.
Data output.	API call parameters passed by reference for cryptographic service output.

The Control Output logical interface is not applicable to the Module and is intentionally omitted from Table 6.

4 Roles, Services, and Authentication

The Module supports the mandatory Cryptographic Officer (CO) operational role only (implicitly identified) and does not support a maintenance role or a bypass capability. The Module does not provide an authentication or identification method of its own. The CO role is assumed by meeting the conditions of Section 11 of this document..

All services implemented by the Module are listed in Table 7, corresponding to the functionality described by the *fips_query* function, which returns available services based on an *operation_id* input.

The *fips_get_params* function provides access to the current status of the Module as well as the name and version; this information correlates to the validation listing. A 1 value returned in status indicates the Module is running without error (FIPS_OK); a 0 return indicates an error (with additional error details indicated as described in the release specific API documentation). Services are only operational in the running state. Any attempts to access services in any other state will result in an error being returned. If the integrity test or any CAST fails then any attempt to access any service will result in an error being returned.

The OpenSSL toolkit *OSSL_PROVIDER_get_params* function is used to invoke *fips_get_params*, when called with the Module's global handle and a pointer to a parameter structure (initialized using *provider_gettable_params* or the equivalent).

Table 7 - Roles, Service Commands, Input and Output

Role	Service	Input	Output
CO	Initialize.	Core handle, dispatch in and out, provider context.	Initialization status (1 = pass, 0 = fail).
Core operations dispatched by the FIPS provider			
CO	Show status, Show module's versioning information Metadata functions fulfill AS04.13 and AS04.14 requirements; see notes below Table 7.	Provider context, parameters types (array).	Parameter types (array) with: Name, Version, BuildInfo, Status, SecurityChecks; Status return.
CO	Get capabilities.	Provider context, capability, callback pointer and arguments.	TLS group capabilities.
CO	Query (available crypto operations).	Provider context, operation ID.	Null or array of available operations.
CO	Perform self-tests.	Provider context.	Status (1 = pass, 0 = fail).
CO	Teardown (Perform zeroisation).	Provider context.	None.
Cryptographic implementation operations			
CO	Asymmetric cipher (Key transport) OSSL_OP_ASYM_CIPHER, OSSL_OP_KEM (Perform approved security functions).	Encapsulate: Key struct (KTS_KDK); Decapsulate (KTS_KEK).	Status return; KTS_SS.
CO	Cipher (Encryption/Decryption and Key Wrapping) OSSL_OP_CIPHER (Perform approved security functions).	SC_EDK; flags.	Status return. Plaintext or ciphertext data.
CO	Key derivation OSSL_OP_KDF (Perform approved security functions).	KAS_SS; flags.	Status return; KD_DKM.
CO	Key exchange (key agreement) OSSL_OP_KEYEXCH (Perform approved security functions).	Key structs (KAS_Private and KAS_Public); flags.	Status return; KAS_SS.

Role	Service	Input	Output
CO	Key management OSSL_OP_KEYMGMT (Perform approved security functions).	ECDSA, EdDSA: curve identifier. DSA/RSA: modulus size.	Status return; Key struct (GKP_Private, GKP_Public).
CO	Message authentication OSSL_OP_MAC (Perform approved security functions).	KH_Key.	Status return; Tag value.
CO	Message digest OSSL_OP_DIGEST (Perform approved security functions).	Message; flags.	Status return; Hash value.
CO	Random OSSL_OP_RAND (Perform approved security functions).	DRBG struct (RBG State); DRBG_EI.	Status return; Random value (DRBG_Output).
CO	Signature OSSL_OP_SIGNATURE (Perform approved security functions).	Sign: Key struct (DS_SGK); message; Verify: signature value; Key struct (DS_SVK); flags; sizes.	Status return; Signature value.
CO	Zeroise: OpenSSL_cleanse (Perform zeroisation).	Memory pointer.	Void.

Table 8 describes Module service access to SSPs; '--' indicates the cell is intentionally empty, not applicable or not relevant. The following annotations indicate the type of access by the Module service:

- **G = Generate**: The Module generates or derives the SSP.
- **R = Read**: The SSP is read from the Module (e.g. the SSP is output).
- **W = Write**: The SSP is updated, imported, or written to the Module.
- **E = Execute**: The Module uses the SSP in performing a cryptographic operation.
- **Z = Zeroise**: The Module zeroises the SSP.

Table 8 - Approved Services

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Initialize.	Module initialization. Does not access CSPs.	DRBG (instantiate).	DRBG_EI DRBG_State.	CO	G/W,EZ G	FIPS_OK.
Core (all except Teardown).	Show status. Core operations dispatched by FIPS provider: Metadata (Gettable parameters; Get parameters; Get capabilities); Query; Self-test.	--	--	CO	--	FIPS_OK.
Core: Perform self-tests.	Run the self-test sequence.	All.	-- [1]	CO	--	FIPS_OK.
Core: Teardown (Perform zeroisation).	Uninstantiate the module; includes Zeroise.	--	All.	CO	Z	FIPS_OK.
Cryptographic implementation operations						

[1] ISO 19790 stipulates that parameters used solely for self-test purposes in 7.10 need not meet zeroisation requirements.

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Asymmetric cipher (Key Transport) (Perform approved security functions).	Encapsulate or decapsulate key material on behalf of the calling process (does not establish keys into the module).	KTS-OAEP-basic (no key confirmation): RSA-OAEP, RSADP, RSAEP.	KTS_KDK KTS_KEK KTS_SS.	CO	E E O	fips=yes.
Cipher (Encryption/Decryption and Key Wrapping) (Perform approved security functions).	Encrypt or decrypt data, including AEAD modes (CCM, GCM) and key wrap (KW, KWP). (CSPs are passed in by the calling process or generated within the module).	CBC, CFB1, CFB8, CFB128, CTR, ECB, OFB [38A] CBC-CS1, CBC-CS2, CBC-CS3 [38A_Add] CCM [38C] CMAC [38B] GCM, GMAC [38D] KW, KWP [38F] XTS [38E].	SC_EDK.	CO	E	fips=yes.
Key derivation (Perform approved security functions).	Derive keying material.	KDA Two Step KDF KDA One-Step KDF ANS 9.42 KDF ANS 9.63 KDF SSH KDF TLS v1.2 KDF TLS v1.3 KDF SP 800-108 KDF Counter, Feedback HKDF PBKDF CKG.	KAS_SS KD_DKM KTS_SS.	CO	E, W G, R E, W	fips=yes.
Key exchange (Perform approved security functions).	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module).	CVL (KAS ECC CDH Component), KAS-FFC-SSC, KAS-ECC-SSC [SP800-56Ar3] KAS-IFC-SSC, [SP800-56Br2].	KAS_Private KAS_Public KAS_SS.	CO	E E G	fips=yes.
Key management (Perform approved security functions).	Generate asymmetric key pairs.	DSA KeyGen, ECDSA KeyGen, RSA KeyGen, EdDSA KeyGen, Safe Prime Groups KeyGen, CKG.	GKP_Private GKP_Public.	CO	G G	fips=yes.

Service	Description	Approved Security Functions	Keys and/or SSPs	Roles	Access Rights to Keys and/or SSPs	Indicator
Message authentication (Perform approved security functions).	Generate or verify data integrity. (CSPs are passed in by the calling process or generated within the module).	CMAC, HMAC, KMAC.	KH_Key.	CO	E	fips=yes.
Message digest (Perform approved security functions).	Generate a message digest.	SHA1; SHA2; SHA3.	--	CO	--	fips=yes.
Random (Perform approved security functions).	Generate random bits using the DRBG.	Hash DRBG, HMAC DRBG, CTR DRBG.	DRBG_EI DRBG_State DRBG_Output.	CO	E E	fips=yes.
Signature (Perform approved security functions).	Generate or verify digital signatures. (SSPs are passed in by the calling process.).	DSA SigGen, ECDSA SigGen, RSA SigGen, CVL (ECDSA SigGen component), CVL (RSASP1).	DS_SGK DS_SVK.	CO	E E	fips=yes.
Zeroise (Perform zeroisation).	The core Teardown operation zeroizes all Module scope SSPs (see above). Call stack cleanup is the duty of the application. Restarting the general-purpose computer clears all SSPs in RAM. *OPENSSL_cleanse* provides zeroisation of SSPs managed by the caller. See the notes below this table for additional explanation.	--	All.	CO	Z	ZERO_OK.

Regarding the Indicator of approved security services, the Module conforms to [FIPS140-3_IG] 2.4.C *Approved Security Service Indicator*, similar to example 2. The Module's name and version parameters (as cited in Section 2) along with the Module's internal indicators of the security-check and conditional-errors settings[2] are used to confirm the Module is the validated Module operating in the approved mode with only approved security services.

Each service provides context sensitive status responses as described in the OpenSSL 3 API manual pages; generally, functions of return type int return the value 1 for success with other error codes as appropriate for the call (described in API documentation).

Note that the caller provides the KAS_Private and KAS_Public keys for shared secret computation; the caller's exchange and assurance of PSPs with the remote participant is outside the scope of the Module.

All CSPs are zeroized (overwritten with 0s) when they are no longer needed:

- Temporary copies of CSPs are zeroised within the relevant function for the scope within which they are used.

[2] Refer to the discussion above on the use of *OSSL_PROVIDER_get_params* to invoke *fips_get_params*.

- CSPs with a lifetime associated with an OpenSSL object (e.g., EVP_PKEY) will be zeroized when freed or reinitialized.
- CSPs with a lifetime associated with the Module are zeroised on Module uninstantiation (the Teardown operation).

The *OPENSSL_cleanse* function is used to zeroise CSPs owned by the caller.

Table 9 – Non-Approved Services

Service	**Description**	**Algorithms Accessed**	**Role**	**Indicator**
Signature.	Generate or verify digital signatures. (SSPs are passed in by the calling process.).	Ed25519, Ed448.	CO.	fips=no.
Key exchange.	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module).	X25519, X448.	CO.	1
Cipher (Encryption/Decryption).	Encrypt or decrypt data (CSPs are passed in by the calling process.).	Triple-DES.	CO.	fips=no.

5 Software/Firmware Security

The Module uses HMAC-SHA2-256 as the approved integrity technique; the file fipsmodule.cnf contains the integrity reference value. The HMAC-SHA2-256 CAST is performed prior to the software integrity test. The Module is provided in an executable form (as fips.so shared object for use in Linux environments and fips.dll for use in Windows environments).

The operator can initiate the integrity test on demand by calling *fips_self_test* (invoked using *OSSL_PROVIDER_self_test* called with the Module's global handle) or reloading the Module. The module does not support loading of any additional software.

In accordance with [ISO19790] Annex B, as the Module is open source, the tools used to build the Module as tested are:

- gcc version 9.3.0
- perl v5.30.0
- gnu make v4.2.1

i. Compilers Used for Each Operational Environment

The specific compilers used to generate the Module for the respective operational environments are listed below:

- Ubuntu Linux 22.04.1 Server: gcc 11.2.0
- Debian 11.5: gcc 10.2.1
- FreeBSD 13.1: clang 11.0.1
- Windows 10: Visual Studio 2019
- macOS 11.5.2 (M1): clang 12.0.5
- macOS 11.5.2 (i7): clang 12.0.5

6 Operational Environment

The operational environment for the Module is modifiable as it runs in General Purpose Computers (GPC).

Table 2 lists the operational environments on which the Module was tested; no operational environment restrictions are required for operation in the approved mode.

All conditions for operation of the Module in the approved mode are given in Section 2.

7 Physical Security

Physical Security requirements are not applicable for this software Module.

8 Non-Invasive Security

In accordance with current CMVP policy, Non-Invasive Security is not applicable.

9 Sensitive Security Parameters (SSPs)

All SSPs used by the Module are described in this section, arranged for consistency with Table 8; ‘--’ indicates the cell is intentionally empty, not applicable, or not relevant.

Keys used for CASTs and the temporary value used in the integrity test are not SSPs; however, the latter is deleted after use as required by AS05.10. Equivalent strength is given for each key or algorithm type (as some algorithms do not use or produce keys).

Table 10 - SSPs

Key/SSP Name/Type	Strength[3]	Security Function and Cert. Number	Generation	Import/Export	Establishment	Storage	Zeroisation	Use & Related Keys
DS_SGK (CSP).	RSA: 112 or 128 DSA: 80, 112 or 128 ECDSA: 112, 128, 192, 256.	RSA SigGen, #A3548 DSA SigGen, #A3548 ECDSA SigGen, #A3548.	--	Call stack (API) input parameters.	--	RAM.	cleared after use.	Key pair for signature generation (using private key DS_SGK) and signature verification (using public key DS_SVK).
DS_SVK (PSP).	RSA: 80[4], 112 or 128 DSA: 80, 112 or 128 ECDSA: 112, 128, 192, 256.	RSA SigVer #A3548 DSA SigVer #A3548 ECDSA SigVer #A3548.	--	Call stack (API) input parameters.	--	RAM.	cleared after use.	Key pair for signature generation (using private key DS_SGK) and signature verification (using public key DS_SVK).
GKP_Private (CSP).	RSA: 112 or 128 DSA: 112 or 128 ECDSA: 112, 128, 192, 256.	RSA KeyGen #A3548 DSA KeyGen #A3548 ECDSA KeyGen #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) output parameters.	--	RAM.	cleared after use.	Key pair (Private: DS_SGK, Public: DS_SVK) generated per caller request; the keypair purpose is unspecified.
GKP_Public (PSP).	RSA: 112 or 128 DSA: 112 or 128 ECDSA: 112, 128, 192, 256.	RSA KeyGen #A3548 DSA KeyGen #A3548 ECDSA KeyGen #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) output parameters.	--	RAM.	cleared after use.	Key pair (Private: GPK_Private, Public: GPK_Public) generated per caller request; the keypair purpose is unspecified.
KAS_Private (CSP).	FFC: between 112 and 200 ECC: 112, 128, 192, 256	KAS #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) input parameters.	--	RAM.	cleared after use.	Key pair component provided by the local participant, used for Diffie-Hellman shared secret generation.

[3] Strength is provided in bits. Please refer to Table 3 and the notes below it for the strength provenance (traceability to applicable standards and special publications).

[4] DS_SVK only.

	IFC [SP800-56Br2]: 112, 128.							
KAS_Public (PSP).	FFC: between 112 and 200 ECC: 112, 128, 192, 256 IFC [SP800-56Br2]: 112, 128.	KAS #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) input parameters.	--	RAM.	cleared after use.	Key pair component provided by the local participant, used for Diffie-Hellman shared secret generation.
KAS_SS (CSP).	FFC: between 112 and 200 ECC: 112, 128, 192, 256 IFC: 112, 128.	KAS-FFC-SSC KAS-ECC-SSC #A3548.	--	--	Established during Key Agreement.	RAM.	cleared after use.	Shared secret calculation; z output value is expected to be used by a KDF.
KD_DKM (CSP).	HMAC PRF: 160, 224, 256, 384, 512.	KDA Two Step KDF #A3548 #A3548 KDA One-Step KDF #A3548 ANS 9.42 KDF #A3548 ANS 9.63 KDF #A3548 SSH KDF #A3548 TLS v1.2 KDF #A3548 TLS v1.3 KDF #A3548 SP 800-108 KDF Counter, Feedback #A3548 HKDF #A3548 PBKDF #A3548 CKG.	--	--	Derived using an approved KDF.	RAM.	cleared after use.	Key Derivation derived keying material.
KH_Key (CSP).	CMAC: 128, 192, 256 GMAC: 128, 192, 256 HMAC: 160, 256, 512. KMAC: 128, 256.	AES #A3548 HMAC #A3548 KMAC #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) input parameters.	--	RAM.	cleared after use.	Keyed Hash key.
KTS_KDK (CSP).	112, 128, 152, 176	RSA RSA-OAEP, RSADP, RSAEP	--	Call stack (API) input parameters.	--	RAM.	cleared after use.	Private (KDK) or public (KEK) component of an RSA key pair used for

		#A3548.						[SP800-56Br2] RSA key transport.
KTS_KEK (PSP).	112, 128, 152, 176	RSA RSA-OAEP, RSADP, RSAEP #A3548.	--	Call stack (API) input parameters.	--	RAM.	cleared after use.	Private (KDK) or public (KEK) component of an RSA key pair used for [SP800-56Br2] RSA key transport.
KTS_SS (CSP).	112, 128, 152, 176	RSA #A3548.	--	Call stack (API) input parameters, Call stack (API) output parameters.	--	RAM.	cleared after use.	The RSA key transport shared secret.
DRBG_EI (CSP).	128 – 256 bits.	DRBG #A3548.	--	Call stack (API) output parameters.	--	RAM.	cleared after use.	Entropy input from an external source used for DRBG seeding.
DRBG_Seed (CSP).	128 – 256 bits.	DRBG #A3548.	Generated internally using the entropy input.	--	--	RAM.	cleared after use.	Seed generated from the entropy input for the DRBG.
DRBG_State (CSP).	Hash DRBG: 160, 224, 256, 384, 512 HMAC DRBG: 160, 224, 256, 384, 512. CTR DRBG: 128, 192, 256.	DRBG #A3548.	DRBG seed material.	--	--	RAM.	cleared after use.	DRBG state. Hash DRBG: V and C. HMAC DRBG: V and Key. CTR DRBG: V and Key.
DRBG_Output (CSP).	128-512 bits.	DRBG #A3548.	DRBG seed material.	Call stack (API) output parameters.	--	RAM.	Cleared after use.	Random bits output from the DRBG for use in SSP generation.
SC_EDK (CSP).	AES: 128, 192, 256 AES CCM: 128, 192, 256 AES GCM: 128, 192, 256 AES XTS: 128, 256.	AES CBC, CFB1, CFB8, CFB128, CTR, ECB, OFB [38A] CBC-CS1, CBC-CS2, CBC-CS3 [38A_Add] CCM [38C] CMAC [38B] GCM, GMAC [38D] KW, KWP [38F] XTS [38E] #A3548 CKG.	Generated internally using DRBG seed material.	Call stack (API) input parameters. Call stack (API) output parameters.	--	RAM.	cleared after use.	AES key used for symmetric encryption and decryption (including use in key wrapping).

The Module maintains only the DRBG CSPs used for key generation as persistent CSPs; these are used exclusively for approved services.

Table 11 - Non-Deterministic Random Number Generation Specification

Entropy Sources	**Minimum Number of Bits of Entropy**	**Details**
Calling application.	128, 192 or 256 bits.	The Module relies on the use of a [SP800-90B] compliant entropy source outside the Module boundary. The calling application is responsible for use of a [SP800-90B] compliant entropy source with at least 256 bits of security strength. Entropy is supplied to the Module via callback functions (see Section 2ii). The callback functions shall return an error if the minimum entropy strength cannot be met.

DRBG outputs are used internally to the Module for asymmetric key pair generation and used by calling applications to generate a random value (potentially for use as a symmetric key).

The Module:

- Produces random values in accordance with [SP800-133r2] Section 4, in that the DRBG output is provided directly as the random output.
- Does not provide any service beyond random value generation for symmetric key generation. SSPs used with symmetric key algorithms are provided by the calling application.
- Produces asymmetric keys in accordance with [SP800-133r2] Section 5, in that all asymmetric keys generated by the module (the Key management service) provide the output of the approved key generation algorithm with no post-processing or manipulation of the generated key pairs. As noted in the previous item, random values used in the asymmetric key generation algorithms are direct outputs of the DRBG. Keys produced by the module use an internal Counter DRBG for which the minimum key size and equivalent security strength is 128 bits.
- Supports direct generation of symmetric keys in accordance with [SP800-133r2] Section 6.1 and symmetric key derivation in accordance with [SP800-133r2] Section 6.2, using the approved and CAVP listed KDF algorithms.

10 Self-tests

Each time the Module is powered up it tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged. The pre-operational self-tests are available on demand by reloading the Module.

On instantiation, the Module performs the pre-operational self-tests and all CASTs listed below. All KATs must complete successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the self-test failure error state. The error state is persistent and no services are available. All attempts to use the Module's services result in the return of a non-zero error code, PROV_R_FIPS_MODULE_IN_ERROR_STATE. To recover from an error state, reload the Module into memory.

The *fips_self_test* function (inclusive of software integrity verification) can also be called on demand, fulfilling AS05.11.

Pre-Operational Self-Tests

- Software Integrity: HMAC-SHA2-256 over the complete module file image.

Conditional Cryptographic Algorithm Self-Tests (CASTs)

- AES ECB: Encryption KAT, ECB mode; 128-bit key.
- AES ECB: Decryption KAT, ECB mode; 128-bit key.
- AES GCM: Authenticated encryption KAT, GCM mode; 256-bit key.
- AES GCM: Authenticated decryption KAT, GCM mode; 256-bit key.
- CTR DRBG: Generate, Reseed, Instantiate functions (per Section 11.3 of [SP800-90Ar1]) for CTR DRBG (AES-128 with derivation function)
- DSA: Signature generation KAT with 2048-bit key, SHA2-384 (using fixed seed DRBG test instance).
- DSA: Signature verification KAT with 2048-bit key, SHA2-384 (using fixed seed DRBG test instance).
- ECDSA: Signature generation KAT with P-224, K-233 and SHA2-512 (using fixed seed DRBG test instance).
- ECDSA: Signature verification KAT with P-224, K-233 and SHA2-512 (using fixed seed DRBG test instance).
- Hash DRBG: Generate, Reseed, Instantiate functions (per Section 11.3 of [SP800-90Ar1]) for Hash DRBG (SHA2-256).
- HMAC DRBG: Generate, Reseed, Instantiate functions (per Section 11.3 of [SP800-90Ar1]) for HMAC DRBG (SHA-1).
- HMAC-SHA2-256: HMAC tag generation KAT using 256-bit key (performed prior to the software integrity test).
- KAS-ECC-SSC: Ephemeral Unified Shared Secret (Z) Computation (per Section 6 of [SP800-56Ar3]), with P-256.
- KAS-FFC-SSC: dhEphem Shared Secret (Z) Computation (per Section 6 of [SP800-56Ar3]); with L = 2048/N = 256.
- KAS-IFC-SSC: IFC Primitive Computation (Scenario 2 of [FIPS140-3_IG] D.F, Section 8.2.2 in [SP800-56Br2]) using 2048-bit modulus.
- KDF SP800-108: Counter Mode (HMAC-SHA2-256).
- KDA: One-step KDF (per Section 4 of [SP800-56Cr2]) and Two-Step KDF (per Section 5 of [SP800-56Cr2]).
- KTS-IFC: Encrypt and Decrypt for Basic, Decrypt for CRT (per [FIPS140-3_IG] D.G and [SP800-56Br2]) using 2048-bit modulus.
- PBKDF: Derivation of the Master Key (MK) (per Section 5.3 of [SP800-132]).
- RSA: Signature generation KAT with 2048-bit key, SHA2-256, PKCS#1.
- RSA: Signature verification KAT with 2048-bit key, SHA2-256, PKCS#1.
- SHA-1: SHA-1 KAT.
- SHA2: SHA2-512 KAT.

- SHA3: SHA3-256 KAT.
- SP 800-135 KDFs: KATs for each of TLS 1.2, SSHv2, ANSI X9.63-2001 and ANSI X9.42-2001 KDFs.
- TLS v1.3 KDF: TLS v1.3 KDF (per Section 7.1 of [RFC8446]).

Conditional Pairwise Consistency Tests (PCTs)

- PCT performed on ECC key pair generation.
- PCT performed on FFC (DSA) key pair generation.
- PCT performed on IFC (RSA) key pair generation.

11 Life-Cycle Assurance

The Module is provided to vendors who integrate it into their product, typically in a manufacturing environment, and is not provided directly to US or Canadian Federal agencies. Adherence to the instructions in this document maintains security throughout the distribution, build, installation and configuration processes.

An authorized Cryptographic Officer is required to perform these steps on each platform where it is intended to be used. The config file output contains information about the Module (such as the self-test status and the Module checksum) and must not be copied from one machine to another.

i. Crypto Officer Guidance

a. Installation and Usage Guidance

The Module is installed as part of the OpenSSL 3.1.2 library. The source distribution package is located at https://www.openssl.org/source/openssl-3.1.2.tar.gz.

The FIPS Provider can be installed on the Tested Configurations listed in Table 2 by performing the following steps:

1. Build and install OpenSSL 3.1.2 to the default location:

The FIPS Provider (i.e., the Module) does not get built and installed automatically. To install the Module automatically during the normal OpenSSL 3.1.2 installation process it must be enabled by configuring OpenSSL using the 'enable-fips' option.

Unix/Linux/macOS:

```
$ ./Configure enable-fips
$ make
$ make install
```

Windows:

```
$ perl Configure enable-fips
$ nmake
$ nmake install
```

The 'install_fips' make target can also be invoked explicitly to install the FIPS Provider independently, without installing the rest of OpenSSL:

```
$ make install_fips
```

Note: The instructions for building and installing OpenSSL 3.1.2 on other platforms can be found in the platform-specific guidance provided in INSTALL.md and README-FIPS.md in the OpenSSL 3.1.2 distribution package. Please see Appendix A for further information on porting the Module to platforms apart from the Tested Configurations in Table 2.

2. Verify the version:

```
$ openssl version -v
```

The Installation of the FIPS Provider that occurs as a result of Step 1 above ensures that the shared library and the configuration file containing information about the Module (e.g., the Module checksum) is copied to its installed location.

To install the FIPS configuration file to a non-default location, this can be achieved by running the 'fipsinstall' command line application manually:

```
$ openssl fipsinstall -pedantic
```

Please see fipsinstall.html /docs/man3.1/man1/openssl-fipsinstall.html for options supported for the 'openssl fipsinstall' command.

Note: The software integrity check (per Section 5 of this document) is performed using HMAC-SHA2-256 on the Module file to validate that the Module has not been modified. The integrity value is compared to a value written to the config file during installation.

b. CVEs

The publication of a CVE does not require immediate re-validation or maintenance in the CMVP process. The module may be updated in the field as needed depending on the severity or consequences of the CVE. The Module will be kept up to date with re-validation and maintenance as required, generally bundling fixes for known CVEs in a next release.

The OpenSSL organization maintains a Vulnerabilities page which describes known vulnerabilities and potential resolution. These are reported to the NVD, where they are independently assessed. The OpenSSL group publishes fixes for these vulnerabilities according to their triage process.

c. Module Sanitization and Destruction

Sanitization is defined in [ISO19790] as "... the process of removing sensitive information (e.g. SSPs, user data, etc.) from the module, so that it may either be distributed to other operators or disposed."

The Module itself does not manage persistent SSPs, authentication data or any user data. The Module may be securely sanitized by deletion of the folder in which the Module was located.

There are no additional procedures required for secure destruction of the Module.

d. Miscellaneous

The module performs run-time checks related to enforcement of security parameters such as the minimum-security strength of keys, valid key sizes, and usage of approved curves. These checks shall not be disabled (by using OPENSSL_NO_FIPS_SECURITYCHECKS or any other method).

Validation of domain parameters prior to generating keys using functions provided by the module is the responsibility of the Cryptographic Officer and not enforced by the module itself.

12 Mitigation of Other Attacks

The Module implements mitigations for some types of attacks using the constant-time implementations and blinding.

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.

Acronyms and Definitions

- AES: Advanced Encryption Standard
- AES-NI: Advanced Encryption Standard New Instructions
- API: Application Programming Interface
- CAVP: Cryptographic Algorithm Validation Program
- CBC: Cipher-Block Chaining
- CCM: Counter with CBC-MAC
- CFB: Cipher Feedback
- CMAC: Cipher-based Message Authentication Code
- CMVP: Cryptographic Module Validation Program
- CO: Cryptographic Officer
- CPU: Central Processing Unit
- CSP: Critical Security Parameter
- CTR: Counter-mode
- CVL: Component Validation List
- DES: Data Encryption Standard
- DH: Diffie-Hellman
- DRBG: Deterministic Random Bit Generator
- DSA: Digital Signature Algorithm
- ECB: Electronic Code Book
- ECC: Elliptic Curve Cryptography
- ECCCDH: ECC Cofactor Diffie-Hellman
- ECDH: Elliptic Curve Diffie-Hellman
- ECDSA: Elliptic Curve Digital Signature Algorithm
- FFC: Finite Field Cryptography
- FIPS: Federal Information Processing Standard
- GCM: Galois/Counter Mode
- GMAC: Galois Message Authentication Code
- HKDF: HMAC-based Extract-and-Expand KDF
- HMAC: Keyed-Hash Message Authentication Code
- IFC: Integer Factorization Cryptography
- IG: Implementation Guidance
- IV: Initialization Vector
- KAS: Key Agreement Scheme
- KAT: Known Answer Test
- KBKDF: Key-Based Key Derivation Function
- KDF: Key Derivation Function
- KMAC: KECCAK Message Authentication Code
- KTS: Key Transport Scheme
- KW: Key Wrap

- KWP: Key Wrap with Padding
- NIST: National Institute of Standards and Technology
- OFB: Output Feedback
- PAA: Processor Algorithm Accelerators
- PBKDF: Password-Based Key Derivation Function
- PCT: Pair-wise Consistency Test
- PRF: Pseudorandom Function
- PSP: Public Security Parameter
- RAM: Random Access Memory
- RNG: Random Number Generator
- RSA: Rivest, Shamir, and Adleman Algorithm
- RSADP: RSA Decryption Primitive
- RSAEP: RSA Encryption Primitive
- SHA: Secure Hash Algorithm
- SHAKE: Secure Hash Algorithm KECCAK
- SHS: Secure Hash Standard
- SSC: Shared Secret Computation
- SSP: Sensitive Security Parameter
- TLS: Transport Layer Security
- XTS: XEX Tweakable Block Cipher with Ciphertext Stealing

References

- [FIPS140-3]: FIPS 140-3, *Security Requirements for Cryptographic Modules*, 3/22/2019
- [SP800-140_DTR]: NIST SP 800-140, *FIPS 140-3 Derived Test Requirements (DTR): CMVP Validation Authority Updates to ISO/IEC 24759*, 3/20/2020
- [SP800-140A]: NIST SP 800-140A, *CMVP Documentation Requirements: CMVP Validation Authority Updates to ISO/IEC 24759*, 3/20/2020
- [SP800-140B]: NIST SP 800-140B, *CMVP Security Policy Requirements: CMVP Validation Authority Updates to ISO/IEC 24759 and ISO/IEC 19790 Annex B*, 3/20/2020
- [SP800-140Cr2]: NIST SP 800-140C Rev. 2, *Cryptographic Module Validation Program (CMVP)-Approved Security Functions: CMVP Validation Authority Updates to ISO/IEC 24759*, 7/25/2023
- [SP800-140Dr2]: NIST SP 800-140D Rev. 2, *Cryptographic Module Validation Program (CMVP)-Approved Sensitive Security Parameter Generation and Establishment Methods: CMVP Validation Authority Updates to ISO/IEC 24759*, 7/25/2023
- [SP800-140F]: NIST SP 800-140F, *CMVP Approved Non-Invasive Attack Mitigation Test Metrics: CMVP Validation Authority Updates to ISO/IEC 24759*, 3/20/2020
- [FIPS140-3_IG]: *Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program*, 11/22/2023
- [ISO19790]: ISO/IEC 19790:2012 *Information technology -- Security techniques -- Security requirements for cryptographic modules*, 11/1/2015
- [ISO24759]: ISO/IEC 24759:2017 *Information technology -- Security techniques -- Test requirements for cryptographic modules*, 3/1/2017
- [SP800-38A]: NIST SP 800-38A, *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, 12/01/2001
- [SP800-38A_Add]: NIST SP 800-38A Addendum, *Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode*, 10/21/2010
- [SP800-38B]: NIST SP 800-38B, *Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication*, 10/06/2016
- [SP800-38C]: NIST SP 800-38C, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, 7/20/2007
- [SP800-38D]: NIST SP 800-38D, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, 11/28/2007
- [SP800-38E]: NIST SP 800-38E, *Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices*, 1/18/2010
- [SP800-38F]: NIST SP 800-38F, *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*, 12/13/2012
- [SP800-56Ar3]: NIST SP 800-56A Rev. 3, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*, 4/16/2018
- [SP800-56Br2]: NIST SP 800-56B Rev. 2, *Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*, 3/21/2019
- [SP800-56Cr2]: NIST SP 800-56C Rev. 2, *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*, 8/18/2020
- [SP800-57P1r5]: NIST SP 800-57 Part 1 Rev. 5, *Recommendation for Key Management: Part 1 – General*, 5/04/2020
- [SP800-90Ar1]: NIST SP 800-90A Rev. 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, 6/24/2015

- [SP800-90B]: NIST SP 800-90B, *Recommendation for the Entropy Sources Used for Random Bit Generation*, 1/10/2018
- [SP800-107r1]: NIST SP 800-107 Rev. 1, *Recommendation for Applications Using Approved Hash Algorithms*, 8/24/2012
- [SP800-108r1]: NIST SP 800-108 Rev. 1, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, 8/17/2022
- [SP800-131Ar2]: NIST SP 800-131A Rev. 2, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*, 3/21/2019
- [SP800-132]: NIST SP 800-132, *Recommendation for Password-Based Key Derivation: Part 1: Storage Applications*, 12/22/2010
- [SP800-133r2]: NIST SP 800-133 Rev. 2, *Recommendation for Cryptographic Key Generation*, 6/04/2020
- [SP800-135r1]: NIST SP 800-135 Rev. 1, *Recommendation for Existing Application-Specific Key Derivation Functions*, 12/23/2011
- [SP800-185]: NIST SP 800-185, *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash*, 12/22/2016
- [SP800-186]: NIST SP 800-186, *Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters*, 2/03/2023
- [FIPS180-4]: FIPS 180-4, *Secure Hash Standard (SHS)*, 8/04/2015
- [FIPS186-4]: FIPS 186-4, *Digital Signature Standard (DSS)*, 7/19/2013
- [FIPS197]: FIPS 197, *Advanced Encryption Standard (AES)*, 5/09/2023
- [FIPS198-1]: FIPS 198-1, *The Keyed Hash Message Authentication Code (HMAC)*, 7/16/2008
- [FIPS202]: FIPS 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, 8/04/2015
- [RFC7627]: IETF RFC 7627, *Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension*, 9/2015
- [RFC7919]: IETF RFC 7919, *Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)*, 8/2016
- [RFC8446]: IETF RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*, 8/2018