Amazon Web Services, Inc.

Amazon Linux 2023 OpenSSL FIPS Provider

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.6
Last update: 2026-07-20

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for the Amazon Linux 2023 OpenSSL FIPS Provider. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

This Non-Proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	1
	Overall Level	1

Table 1: Security Levels

1.3 Additional Information

In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Amazon Linux 2023 OpenSSL FIPS Provider (hereafter referred to as "the module") is defined as a software module in a multi-chip standalone embodiment. It provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module consists of one software component, the "FIPS provider", which implements the FIPS requirements and the cryptographic functionality provided to the operator.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the fips.so shared library, which contains the compiled code implementing the FIPS provider.

Tested Operational Environment's Physical Perimeter (TOEPP):

The TOEPP of the module is defined as the general-purpose computer on which the module is installed.

Figure 1 shows a block diagram that represents the design of the module when the module is operational and providing services to other user space applications. In this diagram, the physical perimeter of the operational environment (a general-purpose computer on which the module is installed) is indicated by a purple dashed line. The cryptographic boundary is represented by the components painted in orange blocks, which consists only of the shared library implementing the FIPS provider (fips.so).

Green lines indicate the flow of data between the cryptographic module and its operator application, through the logical interfaces defined in *Section 3 Cryptographic Module Interfaces.*

Components in white are only included in the diagram for informational purposes. They are not included in the cryptographic boundary (and therefore not part of the module's validation). For example, the kernel is responsible for managing system calls issued by the module itself, as well as other applications using the module for cryptographic services.

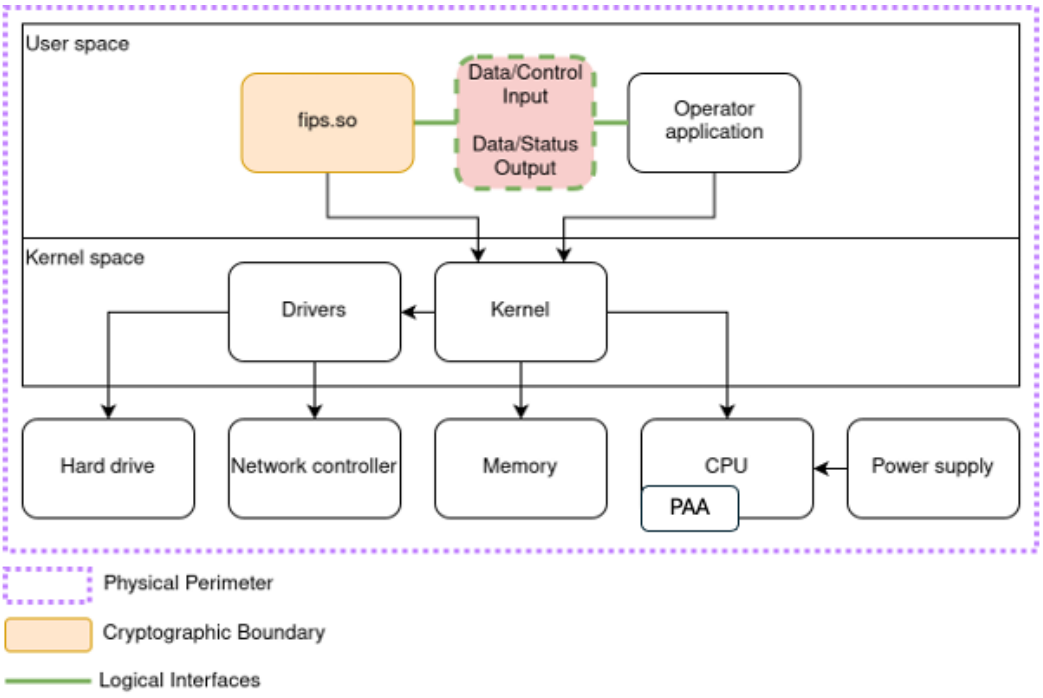

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.so on Amazon Linux 2023 with AWS Graviton3	3.2.2-799901ad7ab41d45	N/A	HMAC-SHA-256
fips.so on Amazon Linux 2023 with Intel Xeon Platinum 8375C	3.2.2-799901ad7ab41d45	N/A	HMAC-SHA-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Amazon Linux 2023	EC2 c7g.metal	AWS Graviton3 (ARMv8-A)	Yes	N/A	3.2.2-799901ad7ab41d45
Amazon Linux 2023	EC2 c6i.metal	Intel Xeon Platinum 8375C (Sunny Cove)	Yes	N/A	3.2.2-799901ad7ab41d45
Amazon Linux 2023	EC2 c7g.metal	AWS Graviton3 (ARMv8-A)	No	N/A	3.2.2-799901ad7ab41d45
Amazon Linux 2023	EC2 c6i.metal	Intel Xeon Platinum 8375C (Sunny Cove)	No	N/A	3.2.2-799901ad7ab41d45

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

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

Operating System	Hardware Platform
Amazon Linux 2023	AWS Snowball with AMD EPYC 9R14

Operating System	Hardware Platform
Amazon Linux 2023	AWS Snowball with AMD EPYC 7702
Amazon Linux 2023	AWS Snowcone with Intel Denverton Atom C3558

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

2.3 Excluded Components

There are no components excluded from the requirements of the FIPS 140-3 standard.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator (specified in Section 4.3) of the requested service
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator (specified in Section 4.3) of the requested service

Table 5: Modes List and Description

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode. No operator intervention is required to reach this point.

In the operational state, the module accepts service requests from calling applications through its logical interfaces. At any point in the operational state, a calling application can end its process, causing the module to end its operation.

Mode Change Instructions and Status:

The module automatically switches between the approved and non-approved modes depending on the services requested by the operator. The status indicator of the mode of operation is equivalent to the indicator of the service that was requested.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A7438, A7442, A7443, A7492, A7493, A7494	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS2	A7438, A7442, A7443, A7492, A7493, A7494	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A7438, A7442, A7443, A7492, A7493, A7494	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A7438, A7442, A7443, A7492, A7493, A7494	Key Length - 128, 192, 256	SP 800-38C

Algorithm	CAVP Cert	Properties	Reference
AES-CFB1	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A7438, A7442, A7443, A7453, A7492, A7493, A7494, A7508, A7509, A7510, A7511	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507	Direction - Decrypt, Encrypt IV Generation - External, Internal Key Length - 128, 192, 256 IV Generation Mode - 8.2.2	SP 800-38D
AES-GMAC	A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-KW	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KWP	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A7438, A7442, A7443, A7492, A7493, A7494	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A7441	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A7445, A7451, A7495, A7496, A7497, A7498	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A7445, A7451, A7495, A7496, A7497, A7498	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7445, A7451, A7495, A7496, A7497, A7498	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Component - No	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7446, A7452	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A7445, A7451, A7495, A7496, A7497, A7498	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A7446, A7452	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
Hash DRBG	A7441	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A7441	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A7444, A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A7445, A7451, A7495, A7496, A7497, A7498	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A7446, A7452	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A7446, A7452	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A7446, A7452	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A7446, A7452	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7445, A7451, A7495, A7496, A7497, A7498	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A7456	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A7440	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A7455	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A7455	MAC Salting Methods - default, random KDF Mode - feedback Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2

Algorithm	CAVP Cert	Properties	Reference
KDF ANS 9.42 (CVL)	A7445, A7451, A7495, A7496, A7497, A7498	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Key Data Length - Key Data Length: 8-4096 Increment 8	SP 800-135 Rev. 1
KDF ANS 9.42 (CVL)	A7446, A7452	KDF Type - DER Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Key Data Length - Key Data Length: 8-4096 Increment 8	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A7445, A7451, A7495, A7496, A7497, A7498	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A7446, A7452	Hash Algorithm - SHA3-224, SHA3-256, SHA3-384, SHA3-512 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-135 Rev. 1
KDF SP800-108	A7454	KDF Mode - Counter, Feedback Supported Lengths - Supported Lengths: 8, 72, 128, 776, 3456, 4096	SP 800-108 Rev. 1
KDF SSH (CVL)	A7453, A7508, A7509, A7510, A7511	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
PBKDF	A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498	Iteration Count - Iteration Count: 1000-10000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-5)	A7445, A7451, A7495, A7496, A7497, A7498	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA SigVer (FIPS186-4)	A7445, A7451, A7495, A7496, A7497, A7498	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024	FIPS 186-4
RSA SigVer (FIPS186-5)	A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
Safe Primes Key Generation	A7456	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A7456	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
SHA-1	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A7444, A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512/256	A7445, A7451, A7495, A7496, A7497, A7498	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A7446, A7452	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A7446, A7452	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A7446, A7452	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A7446, A7452	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A7446, A7452	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A7446, A7452	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A7445, A7451, A7495, A7496, A7497, A7498	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A7440	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms

The table above lists all approved cryptographic algorithms of the module, including specific key lengths employed for approved services (see Approved Services table in *Section 4.3 Approved Services*, and implemented modes or methods of operation of the algorithms.

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Asymmetric Cryptographic Key Generation (CKG)	Key Type:Asymmetric	N/A	SP 800-133Rev2 section 4, example 1

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation with no security claimed.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES GCM (external IV)	Authenticated encryption (not CAVP tested)
HMAC (< 112-bit keys)	Message authentication
KBKDF, KDA OneStep, KDA TwoStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys)	Key derivation
KDA OneStep, KDA TwoStep (SHAKE128, SHAKE256)	Key derivation
ANS X9.42 KDF (SHAKE128, SHAKE256)	Key derivation
ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256)	Key derivation
SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256)	Key derivation
TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3)	Key derivation

Name	Use and Function
TLS 1.2 KDF without Extended Master Secret (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-256, SHA-384, SHA-3)	Key derivation
TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	Key derivation
PBKDF2 (< 8 characters password; < 128 salt length; < 1000 iterations; < 112-bit keys)	Password-based key derivation
RSA (KAS1, KAS2 schemes)	Shared secret computation (not CAVP tested)
RSA and ECDSA (pre-hashed message)	Signature generation; Signature verification
RSA-PSS (invalid salt length)	Signature generation; Signature verification
RSA-OAEP	Asymmetric encryption; Asymmetric decryption (not CAVP tested)

Table 8: Non-Approved, Not Allowed Algorithms

The table above lists all non-approved cryptographic algorithms of the module employed by the non-approved services of the Non-Approved Services table in *Section 4.4 Non-Approved Services.*

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Encryption with AES	BC-UnAuth	Encryption using AES		AES-CBC: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS1: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS2: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS3: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB1: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB128: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB8: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CTR: (A7438, A7442, A7443, A7492, A7493, A7494) AES-ECB: (A7438, A7442, A7443, A7453, A7492, A7493, A7494, A7508, A7509, A7510, A7511) AES-KW: (A7438, A7442, A7443, A7492, A7493, A7494) AES-KWP: (A7438,

Name	Type	Description	Properties	Algorithms
				A7442, A7443, A7492, A7493, A7494) AES-OFB: (A7438, A7442, A7443, A7492, A7493, A7494) AES-XTS Testing Revision 2.0: (A7438, A7442, A7443, A7492, A7493, A7494)
Decryption with AES	BC-UnAuth	Decryption using AES		AES-CBC: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS1: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS2: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CBC-CS3: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB1: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB128: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CFB8: (A7438, A7442, A7443, A7492, A7493, A7494) AES-CTR: (A7438, A7442, A7443, A7492, A7493, A7494) AES-ECB: (A7438, A7442, A7443, A7453, A7492, A7493, A7494, A7508, A7509, A7510, A7511) AES-KW: (A7438, A7442, A7443, A7492, A7493, A7494) AES-KWP: (A7438, A7442, A7443, A7492, A7493, A7494) AES-OFB: (A7438, A7442, A7443, A7492, A7493, A7494) AES-XTS Testing Revision 2.0: (A7438, A7442, A7443, A7492, A7493, A7494)
Authenticated Encryption with AES	BC-Auth	Authenticated encryption using AES		AES-CCM: (A7438, A7442, A7443, A7492,

Name	Type	Description	Properties	Algorithms
				A7493, A7494) AES-GCM: (A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507) AES-KW: (A7438, A7442, A7443, A7492, A7493, A7494) AES-KWP: (A7438, A7442, A7443, A7492, A7493, A7494)
Authenticated Decryption with AES	BC-Auth	Authenticated decryption using AES		AES-CCM: (A7438, A7442, A7443, A7492, A7493, A7494) AES-GCM: (A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507) AES-KW: (A7438, A7442, A7443, A7492, A7493, A7494) AES-KWP: (A7438, A7442, A7443, A7492, A7493, A7494)
KAS-FFC-SSC Shared secret computation	KAS-SSC	Compute shared secret using DH	IG:IG D.F Scenario 2, path (1)	KAS-FFC-SSC Sp800-56Ar3: (A7456)
KAS-ECC-SSC Shared secret computation	KAS-SSC	Compute shared secret using ECDH	IG:IG D.F Scenario 2, path (1)	KAS-ECC-SSC Sp800-56Ar3: (A7445, A7451, A7495, A7496, A7497, A7498)
Hashing	SHA	Compute message digest using SHA		SHA-1: (A7445, A7451, A7495, A7496, A7497, A7498) SHA2-224: (A7445, A7451, A7495, A7496, A7497, A7498) SHA2-256: (A7444, A7445, A7451, A7495, A7496, A7497, A7498) SHA2-384: (A7445, A7451, A7495, A7496, A7497, A7498) SHA2-512: (A7445, A7451, A7495, A7496, A7497, A7498) SHA2-512/224: (A7445, A7451, A7495, A7496, A7497, A7498) SHA2-512/256: (A7445, A7451, A7495, A7496, A7497, A7498)

Name	Type	Description	Properties	Algorithms
				SHA3-224: (A7446, A7452) SHA3-256: (A7446, A7452) SHA3-384: (A7446, A7452) SHA3-512: (A7446, A7452)
MAC Generation with AES-GMAC	MAC	Compute MAC tags using AES-GMAC		AES-GMAC: (A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507)
MAC Verification with AES-GMAC	MAC	Compute MAC tags using AES-GMAC		AES-GMAC: (A7447, A7448, A7449, A7450, A7499, A7500, A7501, A7502, A7503, A7504, A7505, A7506, A7507)
MAC Generation with AES-CMAC or HMAC	MAC	Compute MAC tags using HMAC or AES-CMAC		AES-CMAC: (A7438, A7442, A7443, A7492, A7493, A7494) HMAC-SHA-1: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-224: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-256: (A7444, A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-384: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-512: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-512/224: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA2-512/256: (A7445, A7451, A7495, A7496, A7497, A7498) HMAC-SHA3-224: (A7446, A7452) HMAC-SHA3-256: (A7446, A7452) HMAC-SHA3-384: (A7446, A7452) HMAC-SHA3-512: (A7446, A7452)
Key Pair Generation with RSA	AsymKeyPair-KeyGen CKG	Generate a key pair for RSA		RSA KeyGen (FIPS186-5): (A7445, A7451,

Name	Type	Description	Properties	Algorithms
				A7495, A7496, A7497, A7498)
Key Pair Generation with ECDSA	AsymKeyPair-KeyGen CKG	Generate a key pair for ECDSA		ECDSA KeyGen (FIPS186-5): (A7445, A7451, A7495, A7496, A7497, A7498)
Public Key Verification with ECDSA	AsymKeyPair-KeyVer	Verify public key for ECDSA		ECDSA KeyVer (FIPS186-5): (A7445, A7451, A7495, A7496, A7497, A7498)
Signature Generation with RSA	DigSig-SigGen	Generate a signature using RSA		RSA SigGen (FIPS186-5): (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Signature Verification with RSA	DigSig-SigVer	Verify a signature using RSA		RSA SigVer (FIPS186-5): (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Signature Generation with ECDSA	DigSig-SigGen	Generate a signature using ECDSA		ECDSA SigGen (FIPS186-5): (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Signature Verification with ECDSA	DigSig-SigVer	Verify a signature using ECDSA		ECDSA SigVer (FIPS186-5): (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Extendable Output Function	XOF	Compute message digests from XOFs		SHAKE-128: (A7446, A7452) SHAKE-256: (A7446, A7452)
Random Number Generation with DRBG	DRBG	Generate random numbers using DRBGs		Counter DRBG: (A7441) Hash DRBG: (A7441) HMAC DRBG: (A7441)
Key derivation with KBKDF	KBKDF	Derive keys from a key materials		KDF SP800-108: (A7454)
Key derivation with HKDF	KAS-56CKDF	Derive keys using HKDF		KDA HKDF Sp800-56Cr1: (A7440)
Key derivation with TLS 1.3 KDF	KAS-135KDF	Derive keys from TLS 1.3 KDF		TLS v1.3 KDF: (A7440)
Key derivation with SSH KDF	KAS-135KDF	Derive keys from SSH KDF		KDF SSH: (A7453, A7508, A7509, A7510, A7511)
Key derivation with X9.63 KDF	KAS-135KDF	Derive keys from ANS X9.63 KDF		KDF ANS 9.63: (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Key derivation with X9.42 KDF	KAS-135KDF	Derive keys from ANS X9.42 KDF		KDF ANS 9.42: (A7445, A7446, A7451, A7452,

Name	Type	Description	Properties	Algorithms
				A7495, A7496, A7497, A7498)
Key derivation with PBKDF	PBKDF	Derive keys from PBKDF		PBKDF: (A7445, A7446, A7451, A7452, A7495, A7496, A7497, A7498)
Key Pair Generation with Safe Primes	AsymKeyPair-KeyGen CKG	Generate a key pair from safe primes		Safe Primes Key Generation: (A7456)
Key Pair Verification with Safe Primes	AsymKeyPair-KeyVer	Verify a key pair using safe primes		Safe Primes Key Verification: (A7456)
Key derivation using KDA OneStep	KAS-56CKDF	Key derivation using KDA OneStep		KDA OneStep SP800-56Cr2: (A7455)
Key derivation using KDA TwoStep	KAS-56CKDF	Key derivation using KDA TwoStep		KDA TwoStep SP800-56Cr2: (A7455)
Key derivation with TLS 1.2 KDF	KAS-135KDF	Derive keys from TLS 1.2 KDF		TLS v1.2 KDF RFC7627: (A7445, A7451, A7495, A7496, A7497, A7498)
Signature Verification (Legacy)	DigSig-SigVer	Legacy digital signature verification		RSA SigVer (FIPS186-4): (A7445, A7451, A7495, A7496, A7497, A7498)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

For TLS 1.2, the module offers the AES GCM implementation and uses the context of Scenario 1 of FIPS 140-3 IG C.H. OpenSSL 3 is compliant with SP 800-52r2 Section 3.3.1 and the mechanism for IV generation is compliant with RFC 5288 and 8446.

The module does not implement the TLS protocol. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key. If the counter exhaustion condition is observed, the module returns an error indication to the calling application, which will then need to either abort the connection, or trigger a handshake to establish a new encryption key.

In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES GCM key encryption or decryption under this scenario shall be established.

Alternatively, the Crypto Officer can use the module's API to perform AES GCM encryption using internal IV generation. These IVs are always 96 bits and generated using the approved DRBG internal to the module's boundary, compliant to Scenario 2 of FIPS 140-3 IG C.H.

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. This service can be requested by invoking the EVP_EncryptInit_ex2 API function with a non-NULL iv value. When this is the case, the API will set a non-approved service indicator.

Finally, for TLS 1.3, the AES GCM implementation uses the context of Scenario 5 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher-suites

that explicitly select AES GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES GCM cipher suites from Section 3.3.1 of SP800-52r2. TLS 1.3 employs separate 64-bit sequence numbers, one for protocol records that are received, and one for protocol records that are sent to a peer. These sequence numbers are set at zero at the beginning of a TLS 1.3 connection and each time when the AES-GCM key is changed. After reading or writing a record, the respective sequence number is incremented by one. The protocol specification determines that the sequence number should not wrap, and if this condition is observed, then the protocol implementation must either trigger a re-key of the session (i.e., a new key for AES-GCM) or terminate the connection.

2.7.2 AES XTS

The length of a single data unit encrypted or decrypted with AES XTS shall not exceed 2^{20} AES blocks, that is 16MB, of data per XTS instance. An XTS instance is defined in Section 4 of SP 800-38E.

To meet the requirement stated in IG C.I, the module implements a check that ensures, before performing any cryptographic operation, that the two AES keys used in AES XTS mode are not identical. Key_1 and Key_2 shall be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133r2, Section 6.3.

The XTS mode shall only be used for the cryptographic protection of data on storage devices. It shall not be used for other purposes, such as the encryption of data in transit.

2.7.3 Key Derivation using SP 800-132 PBKDF2

The module provides password-based key derivation (PBKDF2), compliant with SP 800-132. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance to SP 800-132 and FIPS 140-3 IG D.N, the following requirements are met:

- Derived keys shall be used only for storage applications, and shall not be used for any other purposes. The length of the MK or DPK is 112 bits or more (this is verified by the module).
- Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
- The length of the password or passphrase is at least 8 characters (this is verified by the module), and may consist of lowercase, uppercase, and numeric characters. Assuming the worst-case scenario of all digits, the probability is estimated to be at most 10^{-8}. Combined with the minimum iteration count as described in the fifth bullet point, this provides an acceptable trade-off between user experience and security against brute-force attacks.
- A portion of the salt shall be generated randomly using the SP 800-90Ar1 DRBG provided by the module. The minimum length required is 128 bits (this is verified by the module).
- The iteration count is selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value is 1000 (this is verified by the module).

If any of these requirements are not met, the requested service is non-approved (see *Non-Approved Services table* in *Section 4.4 Non-Approved Services*).

2.7.4 SP 800-56Ar3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the operator must use the module together with an application that implements the TLS protocol. Additionally, the module's approved key pair generation service (see *Approved Services table* in *Section 4.3 Approved Services*) must be used to generate ephemeral Diffie-Hellman or EC Diffie-Hellman key pairs, or the key pairs must be obtained from another

FIPS-validated module. As part of this service, the module will internally perform the full public key validation of the generated public key.

The module's shared secret computation service will internally perform the full public key validation of the peer public key, complying with Sections 5.6.2.2.1 and 5.6.2.2.2 of SP 800-56Ar3.

2.7.5 SHA-3

To meet the requirement stated in IG C.C, the module implements the SHA-3 algorithms as both standalone and part of higher-level algorithms. As detailed in *Section 2.6 Security Function Implementations* with corresponding certificates, the cryptographic algorithms that use of SHA-3 include RSA signature generation and verification, ECDSA signature generation and verification, KBKDF, HKDF, X9.63 KDF, X9.42 KDF, PBKDF, OneStep KDA, TwoStep KDA, and HMAC. In addition, the implementation of the extendable output functions SHAKE128 and SHAKE256 were verified to have a standalone usage.

2.7.6 RSA Signatures

To meet the requirement stated in IG C.F, the module implements only the approved modulus sizes of 2048, 3072, and 4096 bits for signature generation. For signature verification, the module implements only the approved module sizes of 1024, 2048, 3072, and 4096 bits. Each algorithm was tested, and corresponding certificates can be found detailed in *Section 2.6 Security Function Implementations.*

2.7.7 Authenticated Encryption/Decryption

The module does not establish SSPs using an approved key transport scheme (KTS). However, it does offer approved authenticated algorithms that can be used by an external operator/application as part of an approved KTS. The authenticated encryption/decryption functions are specified in *Section 2.6 Security Function Implementations.*

2.7.8 KAS-SSC

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS. The KAS-SSC methods are specified in *Section 2.6 Security Function Implementations.*

2.7.9 Legacy Algorithms

Algorithms designated as "Legacy" can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M. Digital signature generation using SHA-1 is non-approved and not allowed in approved services. Digital signature verification using SHA-1 is approved for legacy usage only.

SHA-1 is only approved in the module when used for Message Digest (Hashing), Message Authentication Generation with HMAC, Key derivation with KBKDF, Key derivation with TLS 1.3 KDF, Key derivation with SSH KDF, Key derivation with X9.42 KDF, Key derivation with PBKDF, Key derivation using KDA OneStep, Key derivation using KDA TwoStep and Signature Verification (Legacy). The use of SHA-1 for Digital Signature Generation is non-approved in the module.

The CAVP certificates for the legacy algorithms are listed in the *Approved Algorithms* table.

See section 9.5 regarding algorithm transitions.

2.8 RBG and Entropy

Cert Number	Vendor Name
E125	Amazon Web Services, Inc.

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Amazon OpenSSL CPU Time Jitter RNG Entropy Source	Non-Physical	Amazon Linux 2023 on EC2 c7g.metal with AWS Graviton 3; Amazon Linux 2023 on EC2 c6i.metal with Intel Xeon Platinum 8375C	256 bits	Full entropy	SHA-3 (A4551); HMAC-SHA-512 DRBG (A4551); AES-256 CTR DRBG (A4604)

Table 11: Entropy Sources

The module employs two Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90Ar1. These DRBGs are used internally by the module (e.g. to generate seeds for asymmetric key pairs and random numbers for security functions). They can also be accessed using the specified API functions. The following parameters are used:

1. Private DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate secret random values (e.g. during asymmetric key pair generation). It can be accessed using RAND_priv_bytes.
2. Public DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate general purpose random values that do not need to remain secret (e.g. initialization vectors). It can be accessed using RAND_bytes.

These DRBGs use 384 bits from the entropy source for initial seeding and 256 bits for each subsequent reseed. Outputs of multiple GetEntropy() calls are concatenated to receive the entropy input length greater than 256 bits. The output is truncated to get the entropy input string which is not a multiple of 256. This entropy source is located within the module's physical perimeter but outside of the module's cryptographic boundary. As per the Public document of entropy certificate E125, the entropy source provides full entropy of 256 bits.

2.9 Key Generation

The key generation methods implemented by the module are specified in the Vendor-Affirmed Algorithms table. The key derivation methods implemented by the module are specified in the *Security Function Implementations table*.

Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

2.10 Key Establishment

Key Establishment methods are specified in the *Security Function Implementations table*.

2.11 Industry Protocols

The module implements the SSH key derivation function for use in the SSH protocol (RFC 4253 and RFC 6668).

GCM with internal IV generation in the approved mode is compliant with versions 1.2 and 1.3 of the TLS protocol (RFC 5288 and 8446) and shall only be used in conjunction with the TLS protocol. Additionally, the module implements the TLS 1.2 and TLS 1.3 key derivation functions for use in the TLS protocol.

For Diffie-Hellman, the module supports the use of the following safe primes:

IKE (RFC 3526):

- MODP-2048 (ID = 14)
- MODP-3072 (ID = 15)
- MODP-4096 (ID = 16)
- MODP-6144 (ID = 17)
- MODP-8192 (ID = 18)

TLS (RFC 7919)

- ffdhe2048 (ID = 256)
- ffdhe3072 (ID = 257)
- ffdhe4096 (ID = 258)
- ffdhe6144 (ID = 259)
- ffdhe8192 (ID = 260)

For Elliptic Curve Diffie-Hellman, the module supports the NIST-defined P-224, P-256, P-384, and P-521 curves.

No parts of the SSH, TLS, or IKE protocols, other than those mentioned above, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters
N/A	Data Output	API output parameters
N/A	Control Input	API function calls
N/A	Status Output	API return codes, error queue

Table 12: Ports and Interfaces

The logical interfaces are the APIs through which the applications request services. These logical interfaces are logically separated from each other by the API design. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not support authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 13: Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module when performing a service. The module does not support multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message digest	Compute a message digest	EVP_DigestFinal_ex returns 1	Message	Digest value	Hashing	Crypto Officer
XOF	Compute output of XOF	EVP_DigestFinalXOF_ex returns 1	Message, output length	Digest value	Extendable Output Function	Crypto Officer
Encryption	Encrypt a plaintext	EVP_EncryptFinal_ex returns 1	Plaintext, AES key	Ciphertext	Encryption with AES	Crypto Officer - AES key: W,E
Decryption	Decrypt a plaintext	EVP_DecryptFinal_ex returns 1	Ciphertext, AES key	Plaintext	Decryption with AES	Crypto Officer - AES key: W,E
Authenticated Encryption	Encrypt and authenticate a plaintext	AES GCM: EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_EncryptFinal_ex returns 1	AES key, IV (only CCM and GCM), plaintext	Ciphertext, MAC tag (only CCM and GCM)	Authenticated Encryption with AES	Crypto Officer - AES key: W,E - AES-GCM IV: W,E
Authenticated Decryption	Decrypt and authenticate a ciphertext	AES GCM: EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_DecryptFinal_ex returns 1	AES key, ciphertext, MAC tag (only CCM and GCM), IV (only CCM	Plaintext or failure	Authenticated Decryption with AES	Crypto Officer - AES key: W,E - AES-GCM IV: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			and GCM)			
Message Authentication Code Generation	Computes a MAC tag	HMAC: EVP_MAC_REDHAT_FIPS_INDICATOR_APPROVED; Others: EVP_MAC_final returns 1	Message, AES or HMAC key	MAC tag	MAC Generation with AES-GMAC MAC Verification with AES-GMAC MAC Generation with AES-CMAC or HMAC	Crypto Officer - AES key: W,E - HMAC key: W,E - AES-GCM IV: W,E
Message Authentication Code Verification	Computes a MAC tag	EVP_MAC_final returns 1	Message, AES key	MAC tag	MAC Verification with AES-GMAC	Crypto Officer - AES key: W,E - AES-GCM IV: W,E
Key derivation with KBKDF	Derive a key from a key-derivation key using KBKDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Key-derivation key	KBKDF Derived key	Key derivation with KBKDF	Crypto Officer - Key-derivation key: W,E - KBKDF Derived key: G,R
Key derivation with HKDF	Derive a key from a shared secret using HKDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	HKDF Derived key	Key derivation with HKDF	Crypto Officer - HKDF Derived key: G,R - Shared secret: W,E
Key derivation with TLS KDF	Derive a key from a shared secret using TLS KDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	TLS Derived key	Key derivation with TLS 1.2 KDF Key derivation with TLS 1.3 KDF	Crypto Officer - Shared secret: W,E - TLS Derived key: G,R
Key derivation with SSH KDF	Derive a key from a shared secret using SSH KDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	SSH Derived key	Key derivation with SSH KDF	Crypto Officer - Shared secret: W,E - SSH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Derived key: G,R
Key derivation with X9.63 KDF	Derive a key from a shared secret using X9.63 KDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	X9.63 Derived key	Key derivation with X9.63 KDF	Crypto Officer - Shared secret: W,E - X9.63 Derived key: G,R
Key derivation using X9.42 KDF	Derive a key from a shared secret using X9.42 KDF	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	X9.42 Derived key	Key derivation with X9.42 KDF	Crypto Officer - Shared secret: W,E - X9.42 Derived key: G,R
Key derivation with KDA OneStep	Derive a key from a shared secret using KDA OneStep	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	KDA OneStep Derived key	Key derivation using KDA OneStep	Crypto Officer - KDA OneStep Derived key: G,R - Shared secret: W,E
Key derivation with KDA TwoStep	Derive a key from a shared secret using KDA OneStep	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	KDA TwoStep Derived key	Key derivation using KDA TwoStep	Crypto Officer - KDA TwoStep Derived key: G,R - Shared secret: W,E
Password-based key derivation	Derive a key from a password	EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Password, salt, iteration count	PBKDF Derived key	Key derivation with PBKDF	Crypto Officer - Password: W,E - PBKDF Derived key: G,R
Random number generation	Generate random bytes	EVP_RAND_generate returns 1	Seed, Output length	Random bytes	Random Number Generation with DRBG	Crypto Officer - Entropy input: W,E - DRBG seed: G,E - Internal state (V, Key): G,E - Internal state (V, C): G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Shared secret computation	Compute a shared secret	EVP_PKEY_derive returns 1	DH private key, DH public key; EC private key, EC public key	Shared secret	KAS-FFC-SSC Shared secret computation KAS-ECC-SSC Shared secret computation	Crypto Officer - Shared secret: G,R - DH private key: W,E - DH public key: W,E - EC private key: W,E - EC public key: W,E
Signature generation	Generate a signature	RSA: OSSL_RH_FIPSINDICATOR_APPROVED and EVP_SIGNATURE_REDHAT_FIPS_INDICATOR_APPROVED; ECDSA: OSSL_RH_FIPSINDICATOR_APPROVED	Message, RSA or EC private key	Signature	Signature Generation with RSA Signature Generation with ECDSA	Crypto Officer - RSA private key: W,E - EC private key: W,E
Signature verification	Verify a siganture	RSA: OSSL_RH_FIPSINDICATOR_APPROVED and EVP_SIGNATURE_REDHAT_FIPS_INDICATOR_APPROVED; ECDSA: OSSL_RH_FIPSINDICATOR_APPROVED	Message, signature, RSA or EC public key	Pass/fail	Signature Verification with RSA Signature Verification with ECDSA Signature Verification (Legacy)	Crypto Officer - RSA public key: W,E - EC public key: W,E
Key pair generation	Generate a key pair	EVP_PKEY_generate returns 1	Group; Curve; Modulus bits	DH key pair; EC key pair; RSA key pair	Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes	Crypto Officer - Module Generated DH private key: G,R - Module Generated DH public key: G,R - Module Generated RSA private key: G,R - Module Generated RSA public key: G,R - Module Generated EC private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: G,R - Module Generated EC public key: G,R - Intermediate key generation value: G,E,Z
Public key verification	Verify an EC public key	EVP_PKEY_public_check or EVP_PKEY_private_check or EVP_PKEY_check returns 1	EC public key	Pass/fail	Public Key Verification with ECDSA	Crypto Officer - EC public key: W,E
Key pair verification	Verify a DH key pair	EVP_PKEY_public_check or EVP_PKEY_private_check or EVP_PKEY_check returns 1	DH public key; DH private key	Pass/fail	Key Pair Verification with Safe Primes	Crypto Officer - DH private key: W,E - DH public key: W,E
Show version	Show the name and version information (i.e., via list_provider_info())	None	N/A	Name and version information	None	Unauthenticated
Show status	Return the module status	None	N/A	Module status	None	Unauthenticated
Self-tests	Perform CASTs and integrity test	None	N/A	Pass/fail results of self-tests	Encryption with AES Decryption with AES Authenticated Encryption with AES Authenticated Decryption with AES KAS-FFC-SSC Shared secret computation KAS-ECC-SSC	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Shared secret computation Hashing MAC Generation with AES-CMAC or HMAC MAC Generation with AES-GMAC MAC Verification with AES-GMAC Key Pair Generation with RSA Key Pair Generation with ECDSA Public Key Verification with ECDSA Signature Generation with RSA Signature Verification with RSA Signature Verification (Legacy) Signature Generation with ECDSA Signature Verification with ECDSA	

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Random Number Generation with DRBG Key derivation with KBKDF Key derivation with HKDF Key derivation with TLS 1.2 KDF Key derivation with TLS 1.3 KDF Key derivation with SSH KDF Key derivation with X9.63 KDF Key derivation with X9.42 KDF Key derivation with PBKDF Key Pair Generation with Safe Primes Key derivation using KDA OneStep Key derivation using KDA TwoStep	
Zeroization	Zeroize any SSP	None	Any SSP	None	None	Unauthenticated

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- AES key: Z - HMAC key: Z - Key-derivation key: Z - Shared secret: Z - Password: Z - KBKDF Derived key: Z - HKDF Derived key: Z - X9.63 Derived key: Z - X9.42 Derived key: Z - SSH Derived key: Z - KDA OneStep Derived key: Z - KDA TwoStep Derived key: Z - TLS Derived key: Z - PBKDF Derived key: Z - Entropy input: Z - DRBG seed: Z - Internal state (V, Key): Z - Internal state (V, C): Z - DH private key: Z - DH public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: Z - EC private key: Z - EC public key: Z - RSA private key: Z - RSA public key: Z - Intermediate key generation value: Z

Table 14: Approved Services

The module provides services to operators that assume the available role. All services are described in detail in the API documentation (manual pages). The convention below applies when specifying the access permissions (types) that the service has for each SSP.

- **Generate (G):** The module generates or derives the SSP.
- **Read (R):** The SSP is read from the module (e.g. the SSP is output).
- **Write (W):** The SSP is updated, imported, or written to the module.
- **Execute (E):** The module uses the SSP in performing a cryptographic operation.
- **Zeroize (Z):** The module zeroizes the SSP.
- **N/A:** The module does not access any SSP or key during its operation.

To interact with the module, a calling application must use the EVP API layer provided by OpenSSL. This layer will delegate the request to the FIPS provider, which will in turn perform the requested service. Additionally, this EVP API layer can be used to retrieve the approved service indicator for the module. The redhat_ossl_query_fipsindicator() function indicates whether an EVP API function is approved. After a cryptographic service was performed by the module, the API context (listed in the left column of the bullets below) associated with this request can contain a parameter (listed in the right column of the bullets below) which represents the approved service indicator:

- EVP_CIPHER_CTX: OSSL_CIPHER_PARAM_REDHAT_FIPS_INDICATOR
- EVP_MAC_CTX: OSSL_MAC_PARAM_REDHAT_FIPS_INDICATOR
- EVP_KDF_CTX: OSSL_KDF_PARAM_REDHAT_FIPS_INDICATOR
- EVP_PKEY_CTX: OSSL_KEM_PARAM_REDHAT_FIPS_INDICATOR
- EVP_PKEY_CTX: OSSL_SIGNATURE_PARAM_REDHAT_FIPS_INDICATOR
- EVP_PKEY_CTX: OSSL_ASYM_CIPHER_PARAM_REDHAT_FIPS_INDICATOR

The exact process to use these functions and parameters are described in the module's manual pages.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Encryption	Encrypt a plaintext	AES GCM (external IV)	CO
Message authentication	Compute a MAC tag	HMAC (< 112-bit keys)	CO

Name	Description	Algorithms	Role
Key derivation	Derive a key from a key-derivation key or a shared secret	KBKDF, KDA OneStep, KDA TwoStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys) KDA OneStep, KDA TwoStep (SHAKE128, SHAKE256) ANS X9.42 KDF (SHAKE128, SHAKE256) ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256) SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256) TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3) TLS 1.2 KDF without Extended Master Secret (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-256, SHA-384, SHA-3) TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	CO
Password-based key derivation	Derive a key from a password	PBKDF2 (< 8 characters password; < 128 salt length; < 1000 iterations; < 112-bit keys)	CO
Shared secret computation	Compute a shared secret	RSA (KAS1, KAS2 schemes)	CO
Signature generation	Generate a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO
Signature verification	Verify a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO
Asymmetric encryption	Encrypt a plaintext	RSA-OAEP	CO
Asymmetric decryption	Decrypt a ciphertext	RSA-OAEP	CO

Table 15: Non-Approved Services

The table above lists the non-approved services in this module, the algorithms involved, the roles that can request the service. In this table, CO specifies the Crypto Officer role.

4.5 External Software/Firmware Loaded

The module does not support External Software/Firmware loading.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the fips.so file that was computed at build time. The key used for the HMAC SHA-256 integrity check is embedded in the fips.so file.

5.2 Initiate on Demand

Integrity tests are performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity test may be invoked on-demand by unloading and subsequently re-initializing the module. This will perform (among others) the software integrity test.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

Any SSPs contained within the module are protected by the process isolation and memory separation mechanisms provided by the Linux kernel, and only the module has control over these SSPs.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in *Section 11 Life-Cycle Assurance*. If properly installed, the operating system provides process isolation and memory protection mechanisms that ensure appropriate separation for memory access among the processes on the system. Each process has control over its own data and uncontrolled access to the data of other processes is prevented.

Instrumentation tools like the ptrace system call, gdb and strace, userspace live patching, as well as other tracing mechanisms offered by the Linux environment such as ftrace or systemtap, shall not be used in the operational environment. The use of any of these tools implies that the cryptographic module is running in a non-validated operational environment.

7 Physical Security

The module is comprised of software only, and therefore this section is not applicable.

8 Non-Invasive Security

This module does not implement any non-invasive security mechanism, and therefore this section is not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution. SSPs are stored until they are zeroized by the operator (using a zeroization call or removing power from the module) or zeroized automatically.	Dynamic

Table 16: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input parameters	Operator calling application (TOEPP)	Cryptographic module	Plaintext	Manual	Electronic	
API output parameters	Cryptographic module	Operator calling application (TOEPP)	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the cipher handle	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization procedure succeeded.	By calling the appropriate zeroization functions: AES key: EVP_CIPHER_CTX_free and EVP_MAC_CTX_free; HMAC key: EVP_MAC_CTX_free; Key-derivation key: EVP_KDF_CTX_free; Shared secret: EVP_KDF_CTX_free; Password: EVP_KDF_CTX_free; KBKDF Derived key: EVP_KDF_CTX_free; HKDF Derived key: EVP_KDF_CTX_free; TLS Derived key: EVP_KDF_CTX_free; SSH Derived key: EVP_KDF_CTX_free; X9.63 Derived key: EVP_KDF_CTX_free; X9.42 Derived key: EVP_KDF_CTX_free; PBKDF Derived key: EVP_KDF_CTX_free; KDA OneStep Derived key: EVP_KDF_CTX_free; KDA TwoStep Derived key: EVP_KDF_CTX_free; Entropy input: EVP_RAND_CTX_free; DRBG seed: EVP_RAND_CTX_free; Internal state: EVP_RAND_CTX_free; DH public & private key: EVP_PKEY_free; EC public & private key: EVP_PKEY_free; RSA public & private key: EVP_PKEY_free
Automatic	Automatically zeroized by the	Memory occupied by SSPs is overwritten with zeroes,	N/A

Zeroization Method	Description	Rationale	Operator Initiation
	module when no longer needed	which renders the SSP values irretrievable	
Remove power from the module	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when the module is unloaded. The successful completion of the removal of power from the module indicates that zeroization has completed.	By unloading the module

Table 18: SSP Zeroization Methods

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key used for encryption, decryption, and computing MAC tags	XTS: 256, 512 bits; Other modes: 128, 192, 256 bits - XTS: 128, 256 bits; Other modes: 128, 192, 256 bits	Symmetric key - CSP			Encryption with AES Decryption with AES Authenticated Encryption with AES Authenticated Decryption with AES MAC Generation with AES-CMAC or HMAC
HMAC key	HMAC key used for computing MAC tag	112-524288 bits - 112-256 bits	Symmetric key - CSP			MAC Generation with AES-CMAC or HMAC
Shared secret	Shared secret generated by (EC) Diffie-Hellman	224-8192 bits - 112-256 bits	Shared secret - CSP		KAS-FFC-SSC Shared secret computation KAS-ECC-SSC Shared secret computation	Key derivation with HKDF Key derivation with TLS 1.3 KDF Key derivation with SSH KDF Key derivation with X9.63 KDF Key derivation with X9.42 KDF Key derivation using KDA OneStep Key derivation using KDA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						TwoStep Key derivation with TLS 1.2 KDF
Key-derivation key	Symmetric key used to derive symmetric keys	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key derivation with KBKDF
Password	Password used to derive symmetric keys	8-128 characters - N/A	Password - CSP			Key derivation with PBKDF
KBKDF Derived key	Symmetric key derived from a key-derivation key	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with KBKDF		
HKDF Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with HKDF		
TLS Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with TLS 1.2 KDF Key derivation with TLS 1.3 KDF		
SSH Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with SSH KDF		
X9.63 Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with X9.63 KDF		
X9.42 Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with X9.42 KDF		
PBKDF Derived key	Symmetric key derived from a password	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation with PBKDF		
KDA OneStep Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation using KDA OneStep		
KDA TwoStep Derived key	Symmetric key derived from a shared secret	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key derivation using KDA TwoStep		
Entropy input	Entropy input used to seed the DRBGs	128-384 bits - 128-256 bits	Entropy input - CSP			Random Number Generation with DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG seed	DRBG seed derived from entropy input	CTR_DRBG: 256, 320, 384 bits; Hash_DRBG: 440, 888 bits; HMAC_DRBG: 160, 256, 512 bits - CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRBG: 128, 256 bits	Seed - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
Internal state (V, Key)	Internal state of CTR_DRBG and HMAC_DRBG instances	CTR_DRBG: 256, 320, 348 bits; HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
Internal state (V, C)	Internal state of Hash_DRBG	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
DH private key	Private key used for Diffie-Hellman	2048-8192 bits - 112-200 bits	Private key - CSP			KAS-FFC-SSC Shared secret computation Key Pair Verification with Safe Primes
DH public key	Public key used for Diffie-Hellman	2048-8192 bits - 112-200 bits	Public key - PSP			KAS-FFC-SSC Shared secret computation Key Pair Verification with Safe Primes
EC private key	Private key used for ECDH and ECDSA	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Private key - CSP			KAS-ECC-SSC Shared secret computation Public Key Verification with ECDSA Signature Generation with ECDSA
EC public key	Public key used for ECDH and ECDSA	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Public key - PSP			KAS-ECC-SSC Shared secret computation Public Key Verification with ECDSA Signature Verification with ECDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA private key	Private key used for RSA signature generation	2048-16384 bits - 112-256 bits	Private key - CSP			Signature Generation with RSA
RSA public key	Public key used for RSA signature verification	Signature verification: 1024-16384 bits; Key pair generation: 2048-16384 bits - Signature verification: 80-256 bits; Key pair generation: 112-256 bits	Public key - PSP			Signature Verification with RSA
Module Generated DH private key	Private key used for Diffie-Hellman	2048-8192 bits - 112-200 bits	Private key - CSP	Key Pair Generation with Safe Primes		
Module Generated DH public key	Public key used for Diffie-Hellman	2048-8192 bits - 112-200 bits	Public key - PSP	Key Pair Generation with Safe Primes		
Module Generated EC private key	Private key used for ECDH and ECDSA	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation with ECDSA		
Module Generated EC public key	Public key used for ECDH and ECDSA	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation with ECDSA		
Module Generated RSA private key	Private key used for RSA signature generation	2048-16384 bits - 112-256 bits	Private key - CSP	Key Pair Generation with RSA		
Module Generated RSA public key	Public key used for RSA signature verification	Signature verification: 1024-16384 bits; Key pair generation: 2048-16384 bits - Signature verification: 80-256 bits; Key pair generation: 112-256 bits	Public key - PSP	Key Pair Generation with RSA		
Intermediate key generation value	Temporary value generated during key pair generation services	2048-16384 bits - 112-256 bits	Intermediate value - CSP	Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes		Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes
AES-GCM IV	Initialization Vector used with AES-GCM	96 bits - N/A	Initialization Vector - PSP	Random Number Generation with DRBG		Authenticated Encryption with AES Authenticated Decryption with AES

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	
HMAC key	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	
Shared secret	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	DH private key:Established By DH public key:Established By EC private key:Established By EC public key:Established By HKDF Derived key:Derives KDA OneStep Derived key:Derives KDA TwoStep Derived key:Derives TLS Derived key:Derives SSH Derived key:Derives X9.63 Derived key:Derives X9.42 Derived key:Derives
Key-derivation key	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	KBKDF Derived key:Derives
Password	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	PBKDF Derived key:Derives
KBKDF Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Key-derivation key:Derived From
HKDF Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
TLS Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
SSH Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
X9.63 Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle	Shared secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Remove power from the module	
X9.42 Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
PBKDF Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Password:Derived From
KDA OneStep Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
KDA TwoStep Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Shared secret:Derived From
Entropy input		RAM:Plaintext	From generation until DRBG seed is created	Automatic Remove power from the module	DRBG seed:Derives
DRBG seed		RAM:Plaintext	While the DRBG is instantiated	Automatic Remove power from the module	Entropy input:Derived From Internal state (V, Key):Generates Internal state (V, C):Generates
Internal state (V, Key)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Remove power from the module	DRBG seed:Generated From
Internal state (V, C)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Remove power from the module	DRBG seed:Generated From
DH private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated DH public key:Paired With Intermediate key generation value:Generated From
DH public key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated DH private key:Paired With Intermediate key generation value:Generated From
EC private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated EC public key:Paired With Intermediate key generation value:Generated From
EC public key	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle	Module Generated EC private key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API output parameters			Remove power from the module	Intermediate key generation value:Generated From
RSA private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated RSA public key:Paired With Intermediate key generation value:Generated From
RSA public key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated RSA private key:Paired With Intermediate key generation value:Generated From
Module Generated DH private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated DH public key:Paired With Intermediate key generation value:Generated From
Module Generated DH public key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated DH private key:Paired With Intermediate key generation value:Generated From
Module Generated EC private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated EC public key:Paired With Intermediate key generation value:Generated From
Module Generated EC public key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated EC private key:Paired With Intermediate key generation value:Generated From
Module Generated RSA private key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated RSA public key:Paired With Intermediate key generation value:Generated From
Module Generated RSA public key	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Module Generated RSA private key:Paired With Intermediate key generation value:Generated From
Intermediate key generation value		RAM:Plaintext	For the duration of the service	Automatic	DH private key:Generates DH public key:Generates EC private key:Generates EC public key:Generates RSA private key:Generates RSA public key:Generates
AES-GCM IV	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle	AES key:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API output parameters			Remove power from the module	

Table 20: SSP Table 2

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2031.

The FIPS 186-4, has been superseded by FIPS 186-5. FIPS 186-4 was withdrawn on February 3, 2024. The details regarding usage of Legacy algorithms are specified in Section 2.7.

Keys with security strength less than 128 bits will be non-approved for all purposes starting January 1, 2031.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A7496) - x86	256-bit key	Message authentication	SW/FW Integrity	Module becomes operational	Integrity test for fips.so
HMAC-SHA2-256 (A7451) - arm64	256-bit key	Message authentication	SW/FW Integrity	Module becomes operational	Integrity test for fips.so

Table 21: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is initialized, before the module transitions into the operational state. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A7445)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A7451)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A7495)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A7496)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A7497)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A7498)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A7445)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A7451)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A7495)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A7496)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-512 (A7497)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A7498)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A7446)	SHA3-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A7452)	SHA3-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7444)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7445)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7451)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7495)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7496)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7497)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A7498)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A7446)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A7452)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-GCM (A7447) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7447) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7448) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7448) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A7449) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7449) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7450) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7450) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7499) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7499) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7500) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7500) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7501) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7501) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7502) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7502) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7503) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7503) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7504) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7504) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7505) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A7505) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7506) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7506) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7507) - Encrypt	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A7507) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7438) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7442) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7443) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7453) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7492) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7493) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7494) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7508) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7509) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7510) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A7511) - Decrypt	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
KDF SP800-108 (A7454)	Counter mode; HMAC-SHA-256; 128-bit input key	KAT	CAST	Module becomes operational	Key based key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDA HKDF Sp800-56Cr1 (A7440)	SHA2-224, SHA2-256; 48-bit, 392-bit input secret	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA TwoStep SP800-56Cr2 (A7455)	HMAC-SHA2-256, 48-bit input secret	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
KDA OneStep SP800-56Cr2 (A7455)	SHA-224; 392-bit input secret	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDF SSH (A7453)	SHA-1; 1056-bit input secret	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A7508)	SHA-1; 1056-bit input secret	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A7509)	SHA-1; 1056-bit input secret	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A7510)	SHA-1; 1056-bit input secret	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A7511)	SHA-1; 1056-bit input secret	KAT	CAST	Module becomes operational	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7445)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7451)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7495)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7496)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7497)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A7498)	SHA-256; 384-bit input secret	KAT	CAST	Module becomes operational	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.3 KDF (A7440)	Extract and expand modes; SHA-256	KAT	CAST	Module becomes operational	Industry-based TLS v1.3 KDF key derivation	Test runs at power-on before the integrity test
PBKDF (A7445)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
PBKDF (A7446)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7451)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7452)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7495)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7496)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7497)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A7498)	SHA-256; 24 character password; 288-bit salt; Iteration count: 4096	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
Counter DRBG (A7441)	AES-128 with prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1 including health test per section 11.3 (instantiate, generate, reseed)	Test runs at power-on before the integrity test
Hash DRBG (A7441)	SHA-256 with prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1 including health test per section 11.3 (instantiate, generate, reseed)	Test runs at power-on before the integrity test
HMAC DRBG (A7441)	HMAC-SHA-256 with prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1 including health test per section 11.3 (instantiate, generate, reseed)	Test runs at power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A7456)	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7445)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7451)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 (A7495)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7496)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7497)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7498)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
RSA KeyGen (FIPS186-5) (A7445)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A7451)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A7495)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A7496)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A7497)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A7498)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA SigGen (FIPS186-5) (A7445)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7446)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7451)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7452)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7495)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7496)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A7497)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigGen (FIPS186-5) (A7498)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7445)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7446)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7451)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7452)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7495)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7496)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7497)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A7498)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7445)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7451)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7495)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7496)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7497)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A7498)	PKCS#1 v1.5 with SHA-256; 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-5) (A7445)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5) (A7451)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA KeyGen (FIPS186-5) (A7495)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5) (A7496)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5) (A7497)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5) (A7498)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification; SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
ECDSA SigGen (FIPS186-5) (A7445)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7446)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7451)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7452)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7495)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7496)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7497)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A7498)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7445)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7446)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7451)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7452)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7495)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigVer (FIPS186-5) (A7496)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7497)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A7498)	SHA-256; P-224, P-256, P-384, P-521	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
Safe Primes Key Generation (A7456)	N/A	PCT	PCT	Successful key pair generation	SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
KDF ANS 9.63 (A7445)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7446)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7451)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7452)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7495)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7496)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7497)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A7498)	SHA-256; 192-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7445)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7446)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7451)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7452)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7495)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF ANS 9.42 (A7496)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7497)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A7498)	SHA-1 with AES-128 KW; 160-bit input secret	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test

Table 22: Conditional Self-Tests

Data output through the data output interface is inhibited during the conditional self-tests. The module does not return control to the calling application until the tests are completed. If any of these tests fails, the module transitions to the error state *(Section 10.4 Error States).*

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A7496) - x86	Message authentication	SW/FW Integrity	On demand	Unload and re-initialize the module
HMAC-SHA2-256 (A7451) - arm64	Message authentication	SW/FW Integrity	On demand	Unload and re-initialize the module

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA-1 (A7445)	KAT	CAST	On Demand	Manually
SHA-1 (A7451)	KAT	CAST	On Demand	Manually
SHA-1 (A7495)	KAT	CAST	On Demand	Manually
SHA-1 (A7496)	KAT	CAST	On Demand	Manually
SHA-1 (A7497)	KAT	CAST	On Demand	Manually
SHA-1 (A7498)	KAT	CAST	On Demand	Manually
SHA2-512 (A7445)	KAT	CAST	On Demand	Manually
SHA2-512 (A7451)	KAT	CAST	On Demand	Manually
SHA2-512 (A7495)	KAT	CAST	On Demand	Manually
SHA2-512 (A7496)	KAT	CAST	On Demand	Manually
SHA2-512 (A7497)	KAT	CAST	On Demand	Manually
SHA2-512 (A7498)	KAT	CAST	On Demand	Manually
SHA3-256 (A7446)	KAT	CAST	On Demand	Manually
SHA3-256 (A7452)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7444)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7445)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7451)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7495)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7496)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7497)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A7498)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A7446)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A7452)	KAT	CAST	On Demand	Manually
AES-GCM (A7447) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7447) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7448) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7448) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7449) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7449) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7450) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7450) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7499) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7499) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7500) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7500) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7501) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7501) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7502) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7502) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7503) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7503) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7504) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7504) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7505) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7505) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7506) - Encrypt	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A7506) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7507) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A7507) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7438) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7442) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7443) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7453) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7492) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7493) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7494) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7508) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7509) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7510) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A7511) - Decrypt	KAT	CAST	On Demand	Manually
KDF SP800-108 (A7454)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A7440)	KAT	CAST	On Demand	Manually
KDA TwoStep SP800-56Cr2 (A7455)	KAT	CAST	On Demand	Manually
KDA OneStep SP800-56Cr2 (A7455)	KAT	CAST	On Demand	Manually
KDF SSH (A7453)	KAT	CAST	On Demand	Manually
KDF SSH (A7508)	KAT	CAST	On Demand	Manually
KDF SSH (A7509)	KAT	CAST	On Demand	Manually
KDF SSH (A7510)	KAT	CAST	On Demand	Manually
KDF SSH (A7511)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7445)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7451)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7495)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7496)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7497)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
TLS v1.2 KDF RFC7627 (A7498)	KAT	CAST	On Demand	Manually
TLS v1.3 KDF (A7440)	KAT	CAST	On Demand	Manually
PBKDF (A7445)	KAT	CAST	On Demand	Manually
PBKDF (A7446)	KAT	CAST	On Demand	Manually
PBKDF (A7451)	KAT	CAST	On Demand	Manually
PBKDF (A7452)	KAT	CAST	On Demand	Manually
PBKDF (A7495)	KAT	CAST	On Demand	Manually
PBKDF (A7496)	KAT	CAST	On Demand	Manually
PBKDF (A7497)	KAT	CAST	On Demand	Manually
PBKDF (A7498)	KAT	CAST	On Demand	Manually
Counter DRBG (A7441)	KAT	CAST	On Demand	Manually
Hash DRBG (A7441)	KAT	CAST	On Demand	Manually
HMAC DRBG (A7441)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A7456)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7445)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7451)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7495)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7496)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7497)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7498)	KAT	CAST	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7445)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7451)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7495)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7496)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7497)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A7498)	PCT	PCT	On Demand	Manually
RSA SigGen (FIPS186-5) (A7445)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7446)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7451)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7452)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7495)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7496)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigGen (FIPS186-5) (A7497)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7498)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7445)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7446)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7451)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7452)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7495)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7496)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7497)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7498)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7445)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7451)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7495)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7496)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7497)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A7498)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7445)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7451)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7495)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7496)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7497)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A7498)	PCT	PCT	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7445)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7446)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7451)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7452)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigGen (FIPS186-5) (A7495)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7496)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7497)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7498)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7445)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7446)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7451)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7452)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7495)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7496)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7497)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7498)	KAT	CAST	On Demand	Manually
Safe Primes Key Generation (A7456)	PCT	PCT	On Demand	Manually
KDF ANS 9.63 (A7445)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7446)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7451)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7452)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7495)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7496)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7497)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A7498)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7445)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7446)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7451)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7452)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7495)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7496)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7497)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A7498)	KAT	CAST	On Demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The module immediately stops functioning	Software integrity test failure CAST failure PCT failure	Re-initialization of the module	Module will not load on integrity or CAST failures; Module is aborted for PCT failure

Table 25: Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process. Consequently, the data output interface is inhibited, and the module no longer accepts inputs or requests (as the module is no longer running).

10.5 Operator Initiation of Self-Tests

The software integrity tests and cryptographic algorithm self-tests can be invoked on demand by unloading and subsequently re-initializing the module. The pair-wise consistency tests can be invoked on demand by requesting the key pair generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the Amazon Linux 2023 packages in the form of the openssl-3.2.2-1.amzn2023.0.1 RPM package.

Before the openssl-3.2.2-1.amzn2023.0.1 RPM package is installed, the Amazon Linux 2023 systems must operate in the FIPS validated configuration. This can be achieved by switching the system into the FIPS validated configuration after the installation. Execute the openssl list -providers command. Restart the system. More information can be found at the vendor documentation.

The Crypto Officer must verify the Amazon Linux 2023 systems operates in the FIPS validated configuration by executing the fips-mode-setup –check command, which should output “FIPS mode is enabled.”

11.2 Administrator Guidance

After installation of the openssl-3.2.2-1.amzn2023.0.1 RPM package, the Crypto Officer must verify the module name and version by executing the openssl list -providers command. The Crypto Officer must ensure that the fips provider is listed in the output as follows:

fips
name: Amazon Linux 2023 - OpenSSL FIPS Provider
version: 3.2.2-799901ad7ab41d45
status: active

11.3 Non-Administrator Guidance

There is no administrator guidance.

11.4 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the openssl-3.2.2-1.amzn2023.0.1 RPM package can be uninstalled from the Amazon Linux 2023 systems.

12 Mitigation of Other Attacks

12.1 Attack List

Certain cryptographic subroutines and algorithms are vulnerable to timing analysis. The module mitigates this vulnerability by using constant-time implementations. This includes, but is not limited to:

- Big number operations: computing GCDs, modular inversion, multiplication, division, and modular exponentiation (using Montgomery multiplication)
- Elliptic curve point arithmetic: addition and multiplication (using the Montgomery ladder)
- Vector-based AES implementations.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CE	Cryptography Extensions
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPACF	CP Assist for Cryptographic Functions
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EVP	Envelope
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
KAS	Key Agreement Scheme
KAT	Known Answer Test

KBKDF	Key-based Key Derivation Function
KW	Key Wrap
KWP	Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing