



Amazon Web Services Inc.

AWS-LC Cryptographic Module (dynamic)

FIPS 140-3 Non-Proprietary Security Policy

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

1 General	5
1.1 Overview	5
1.2 Security Levels	5
1.3 Additional Information	5
2 Cryptographic Module Specification	6
2.1 Description	6
2.2 Tested and Vendor Affirmed Module Version and Identification	7
2.3 Excluded Components	8
2.4 Modes of Operation	8
2.5 Algorithms	8
2.6 Security Function Implementations	13
2.7 Algorithm Specific Information	18
2.8 RBG and Entropy	21
2.9 Key Generation	21
2.10 Key Establishment	21
2.11 Industry Protocols	21
3 Cryptographic Module Interfaces	22
3.1 Ports and Interfaces	22
4 Roles, Services, and Authentication	23
4.1 Authentication Methods	23
4.2 Roles	23
4.3 Approved Services	23
4.4 Non-Approved Services	29
4.5 External Software/Firmware Loaded	30
5 Software/Firmware Security	31
5.1 Integrity Techniques	31
5.2 Initiate on Demand	31
6 Operational Environment	32
6.1 Operational Environment Type and Requirements	32
6.2 Configuration Settings and Restrictions	32
7 Physical Security	33
8 Non-Invasive Security	34
9 Sensitive Security Parameters Management	35
9.1 Storage Areas	35
9.2 SSP Input-Output Methods	35
9.3 SSP Zeroization Methods	35
9.4 SSPs	36
9.5 Transitions	40
10 Self-Tests	41

10.1 Pre-Operational Self-Tests	41
10.2 Conditional Self-Tests.....	41
10.3 Periodic Self-Test Information	50
10.4 Error States	56
10.5 Operator Initiation of Self-Tests.....	57
11 Life-Cycle Assurance	58
11.1 Installation, Initialization, and Startup Procedures	58
11.2 Administrator Guidance.....	59
12 Mitigation of Other Attacks	60
12.1 Attack List	60
12.2 Mitigation Effectiveness	60
Appendix A. Glossary and Abbreviations.....	61
Appendix B. References	62

List of Tables

Table 1: Security Levels	5
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets).....	7
Table 3: Tested Operational Environments - Software, Firmware, Hybrid	8
Table 4: Modes List and Description.....	8
Table 5: Approved Algorithms.....	12
Table 6: Vendor-Affirmed Algorithms	12
Table 7: Non-Approved, Allowed Algorithms with No Security Claimed.....	13
Table 8: Non-Approved, Not Allowed Algorithms.....	13
Table 9: Security Function Implementations	18
Table 10: Ports and Interfaces.....	22
Table 11: Roles	23
Table 12: Approved Services.....	29
Table 13: Non-Approved Services	30
Table 14: Storage Areas.....	35
Table 15: SSP Input-Output Methods.....	35
Table 16: SSP Zeroization Methods	35
Table 17: SSP Table 1	38
Table 18: SSP Table 2	40
Table 19: Pre-Operational Self-Tests	41
Table 20: Conditional Self-Tests.....	49
Table 21: Pre-Operational Periodic Information.....	50
Table 22: Conditional Periodic Information.....	56
Table 23: Error States	57

List of Figures

Figure 1: Block Diagram.....	6
------------------------------	---

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version AWS-LC FIPS 2.0.0 of the AWS-LC Cryptographic Module (dynamic). 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.

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

This Security Policy describes the features and design of the module named AWS-LC Cryptographic Module (dynamic) using the terminology contained in the FIPS 140-3 specification. The FIPS 140-3 Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-3. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

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.

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 AWS-LC Cryptographic Module (dynamic) (hereafter referred to as “the module”) provides cryptographic services to applications running in the user space of the underlying operating system through a C language Application Program Interface (API).

Module Type: Software

Module Embodiment: Multi-Chip Standalone

Cryptographic Boundary:

The block diagram in Figure 1 shows the cryptographic boundary of the module and its interfaces with the operational environment and the PAA and the flow of information between the module and operator (depicted through the arrows).

The cryptographic boundary is defined as the bcm.o file (version AWS-LC FIPS 2.0.0). This file is dynamically linked to the userspace application during the compilation process.

Tested Operational Environment’s Physical Perimeter (TOEPP):

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

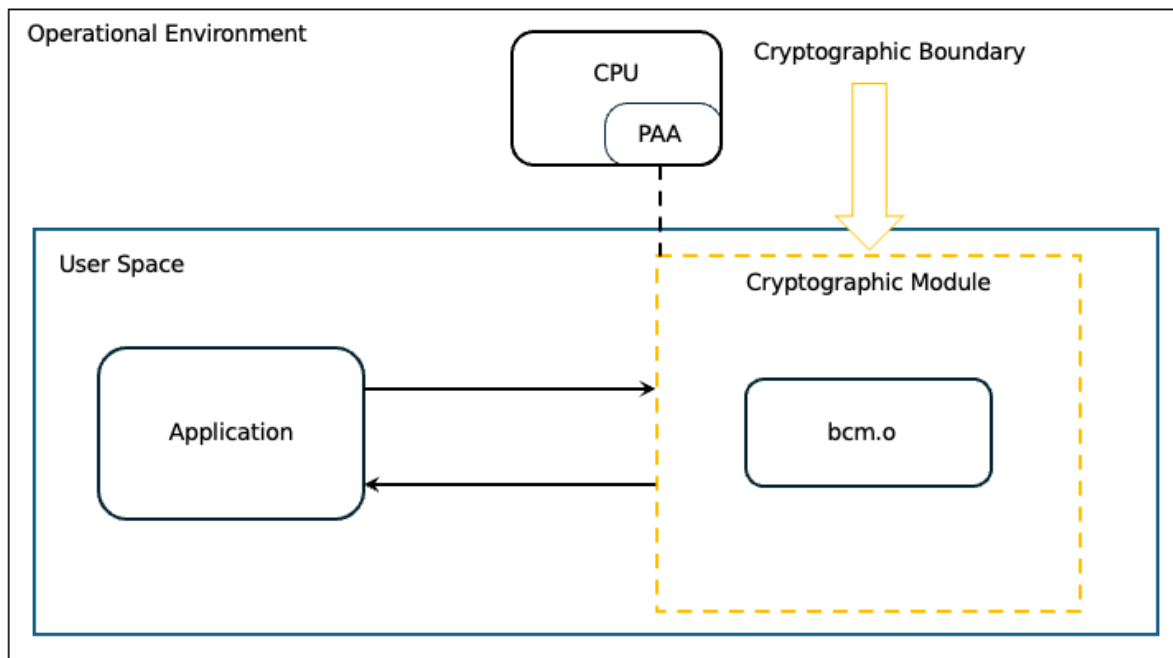


Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification - Hardware:

N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
bcm.o	AWS-LC FIPS 2.0.0		HMAC-SHA2-256

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

Tested Module Identification - Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Amazon Linux 2	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	Yes	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	No	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2023	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	Yes	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2023	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	No	N/A	AWS-LC FIPS 2.0.0
Ubuntu 22.04	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	Yes	N/A	AWS-LC FIPS 2.0.0
Ubuntu 22.04	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel(R) Xeon(R) Platinum 8275CL	No	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	Yes	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	No	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2023	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	Yes	N/A	AWS-LC FIPS 2.0.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Amazon Linux 2023	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	No	N/A	AWS-LC FIPS 2.0.0
Ubuntu 22.04	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	Yes	N/A	AWS-LC FIPS 2.0.0
Ubuntu 22.04	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	No	N/A	AWS-LC FIPS 2.0.0

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

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

N/A for this module.

2.3 Excluded Components

The module does not claim any excluded components.

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	Return value 1 from the API call sequence specified in section 4.3.
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Return value 1 from the API call sequence specified in section 4.3.

Table 4: Modes List and Description

Mode Change Instructions and Status:

When the module starts up successfully, after passing the pre-operational self-test and the cryptographic algorithms self-tests (CASTs), the module is operating in the approved mode of operation by default and can only be transitioned into the non-approved mode by calling one of the non-approved services listed in the Non-Approved Services table. The module will transition back to approved mode when approved service is called. Section 4 provides details on the service indicator implemented by the module. The service indicator identifies when an approved service is called.

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A4484, A4487, A4489, A4493, A4497, A4501	Key Length - 128 Tag Length - 32 IV Length - IV Length: 104 Payload Length - Payload Length: 0-256 Increment 8 AAD Length - AAD Length: 0, 256, 65536	SP 800-38C
AES-CMAC	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Generation, Verification Key Length - 128, 256 MAC Length - MAC Length: 32-128 Increment 8 Message Length - Message Length: 0-65536 Increment 8	SP 800-38B
AES-CTR	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Decrypt, Encrypt Key Length - 128, 192, 256 Payload Length - Payload Length: 8-128 Increment 8 Supports Counter larger than maximum value - Yes Incremental Counter - Yes Counter Tests Performed - Yes	SP 800-38A
AES-ECB	A4484, A4485, A4486, A4487, A4488, A4489, A4490, A4493, A4494, A4495, A4496, A4497, A4498, A4499, A4500, A4501, A4502, A4503, A4504	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A4485, A4486, A4488, A4490, A4494, A4495, A4496, A4498, A4499, A4500, A4502, A4503, A4504	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1, 8.2.2 Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 Payload Length - Payload Length: 128-65536 Increment 128 AAD Length - AAD Length: 0-320 Increment 8	SP 800-38D
AES-GMAC	A4485, A4486, A4488, A4490, A4494, A4495, A4496, A4498, A4499, A4500, A4502, A4503, A4504	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1, 8.2.2 Key Length - 128, 256 Tag Length - 104, 112, 120, 128, 32, 64, 96 IV Length - IV Length: 96 AAD Length - AAD Length: 0-320 Increment 8	SP 800-38D
AES-KW	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256 Payload Length - Payload Length: 128-4096 Increment 128	SP 800-38F
AES-KWP	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Decrypt, Encrypt Cipher - Cipher Key Length - 128, 192, 256	SP 800-38F

Algorithm	CAVP Cert	Properties	Reference
		Payload Length - Payload Length: 8-4096 Increment 8	
AES-XTS Testing Revision 2.0	A4484, A4487, A4489, A4493, A4497, A4501	Direction - Decrypt, Encrypt Key Length - 256 Payload Length - Payload Length: 128-65536 Increment 128 Tweak Mode - Hex Data Unit Length Matches Payload Length - Yes	SP 800-38E
Counter DRBG	A4484, A4487, A4489, A4493, A4497, A4501	Prediction Resistance - No Supports Reseed - No Mode - AES-256 Derivation Function Enabled - No Additional Input - Additional Input: 256 Entropy Input - Entropy Input: 384 Nonce - Nonce: 0 Personalization String Length - Personalization String Length: 384 Returned Bits - 512	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 160 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 224 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 384 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 512 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	MAC - MAC: 256 Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA HKDF Sp800-56Cr1	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Fixed Info Pattern - uPartyInfo vPartyInfo Fixed Info Encoding - concatenation 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	SP 800-56C Rev. 2
KDF SSH (CVL)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KDF TLS (CVL)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	TLS Version - v1.0/1.1, v1.2 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
PBKDF	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Iteration Count - Iteration Count: 1000-10000 Increment 1 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 Password Length - Password Length: 14-128 Increment 1 Salt Length - Salt Length: 128-4096 Increment 8 Key Data Length - Key Data Length: 128-4096 Increment 8	SP 800-132
RSA KeyGen (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Generation Mode - probable Modulo - 2048, 3072, 4096 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Fixed Public Exponent - 010001 Info Generated By Server - No Private Key Format - standard Public Exponent Mode - fixed	FIPS 186-5
RSA SigGen (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-4)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Signature Type - PKCS 1.5, PKCS PSS Modulo - 1024, 2048, 3072, 4096 Hash Pair - Hash Algorithm - SHA2-224 Salt Length - 28 Public Exponent Mode - Fixed Fixed Public Exponent - 010001	FIPS 186-4
RSA SigVer (FIPS186-5)	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Mask Function - mgf1 Fixed Public Exponent - 010001 Public Exponent Mode - fixed	FIPS 186-5
SHA-1	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Asymmetric Cryptographic Key Generation (CKG)	Key Type:Asymmetric	N/A	SP 800-133r2, section 4, direct DRBG output without XOR

Table 6: 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:

Name	Caveat	Use and Function
MD5	Allowed per IG 2.4.A	Message Digest used in TLS 1.0/1.1 KDF only

Table 7: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES with OFB or CFB1, CFB8 modes	Encryption, Decryption (not CAVP tested)
AES-GCM, CCM, GMAC, XTS keys not listed in the Approved Services table of the Security Policy	Encryption, Decryption
AES using aes_*_generic function	Encryption, Decryption (not CAVP tested)
AES GMAC using aes_*_generic	Message Authentication Generation (not CAVP tested)
Curve secp256k1	Signature Generation, Signature Verification, Shared Secret Computation
Diffie Hellman	Shared Secret Computation (not CAVP tested)
HMAC-MD4, HMAC-MD5, HMAC-SHA1, HMAC-SHA-3, HMAC- RIPEMD-160	Message Authentication Generation (not CAVP tested)
MD4	Message Digest
MD5	Message Digest (outside of TLS)
RSA using RSA_generate_key_ex	Key Generation (not complaint with FIPS186-5)
ECDSA using EC_KEY_generate_key	Key Generation (not complaint with FIPS186-5)
RSA using keys less than 2048 bits	Signature Generation
RSA using keys less than 1024 bits	Signature Verification
RSA	Key Encapsulation/Un-encapsulation (not compliant with SP 800-56BRev2), sign/verify primitive operations without hashing
RSA with PKCS#1 v1.5 and OAEP padding	Encryption primitive
SHA-1, SHA-3	Signature Generation (not CAVP tested)
SHAKE, RIPEMD-160, SHA-3	Message Digest (not CAVP tested)
TLS KDF using any SHA algorithms not listed in SP the Approved Services table of the Security Policy or TLS KDF using non extended master secret	Key Derivation

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
ECDSA Key Generation	AsymKeyPair- KeyGen CKG	Key Generation		ECDSA KeyGen (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507) Asymmetric

Name	Type	Description	Properties	Algorithms
				Cryptographic Key Generation (CKG): () Key Type: Asymmetric
RSA Key Generation	AsymKeyPair-KeyGen CKG	Key Generation		RSA KeyGen (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507) Asymmetric Cryptographic Key Generation (CKG): () Key Type: Asymmetric
ECDSA Key Verification	AsymKeyPair-KeyVer	Key Verification		ECDSA KeyVer (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
ECDSA Signature Generation	DigSig-SigGen	Signature Generation (FIPS 186-5)		ECDSA SigGen (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
RSA Signature Generation	DigSig-SigGen	Signature Generation		RSA SigGen (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
ECDSA Signature Verification	DigSig-SigVer	Signature Verification (FIPS 186-5)		ECDSA SigVer (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Legacy ECDSA Signature Verification	DigSig-SigVer	Signature Verification (FIPS 186-4)		ECDSA SigVer (FIPS186-4): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Legacy RSA Signature Verification	DigSig-SigVer	Signature Verification (FIPS 186-4)		RSA SigVer (FIPS186-4): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
RSA Signature Verification	DigSig-SigVer	Signature Verification (FIPS 186-5, FIPS 186-4)		RSA SigVer (FIPS186-4): (A4483, A4492,

Name	Type	Description	Properties	Algorithms
				A4505, A4491, A4508, A4506, A4507) RSA SigVer (FIPS186-5): (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Message Authentication Generation with AES	MAC	Message Authentication Generation		AES-CMAC: (A4501, A4493, A4484, A4489, A4497, A4487) AES-GMAC: (A4500, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4490, A4503)
Message Authentication Generation with HMAC	MAC	Message Authentication Generation		HMAC-SHA-1: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) HMAC-SHA2-224: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) HMAC-SHA2-256: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) HMAC-SHA2-384: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) HMAC-SHA2-512: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) HMAC-SHA2-512/256: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Message Authentication Verification with AES-GMAC	MAC	Message Authentication Generation		AES-GMAC: (A4500, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4490, A4503)

Name	Type	Description	Properties	Algorithms
KAS-ECC-SSC (Shared Secret Computation with EC Diffie-Hellman)	KAS-SSC	Shared Secret Computation	IG:IG D.F Scenario 2 path (1)	KAS-ECC-SSC Sp800-56Ar3: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
HKDF Key Derivation	KAS-56CKDF	Key Derivation		KDA HKDF Sp800-56Cr1: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
SSH Key Derivation	KAS-135KDF	Key Derivation		KDF SSH: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
TLS Key Derivation	KAS-135KDF	Key Derivation		KDF TLS: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
PBKDF Key Derivation	PBKDF	Key Derivation		PBKDF: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Message Digest	SHA	Message Digest		SHA-1: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) SHA2-224: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) SHA2-256: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) SHA2-384: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) SHA2-512: (A4483, A4492, A4505, A4491, A4508, A4506, A4507) SHA2-512/256: (A4483, A4492, A4505, A4491, A4508, A4506, A4507)
Authenticated Encryption	BC-Auth	Authenticated Encryption		AES-GCM: (A4500, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4490, A4503) AES-CCM: (A4501, A4493, A4484,

Name	Type	Description	Properties	Algorithms
				A4489, A4497, A4487) AES-KW: (A4501, A4493, A4484, A4489, A4497, A4487) AES-KWP: (A4501, A4493, A4484, A4489, A4497, A4487)
Authenticated Decryption	BC-Auth	Authenticated Decryption		AES-GCM: (A4500, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4490, A4503) AES-CCM: (A4501, A4493, A4484, A4489, A4497, A4487) AES-KW: (A4501, A4493, A4484, A4489, A4497, A4487) AES-KWP: (A4501, A4493, A4484, A4489, A4497, A4487)
Symmetric Encryption	BC-UnAuth	Encrypt a plaintext		AES-CBC: (A4501, A4493, A4484, A4489, A4497, A4487) AES-CTR: (A4501, A4493, A4484, A4489, A4497, A4487) AES-ECB: (A4500, A4501, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4493, A4484, A4489, A4490, A4503, A4497, A4487) AES-XTS Testing Revision 2.0: (A4501, A4493, A4484, A4489, A4497, A4487)
Symmetric Decryption	BC-UnAuth	Decrypt a ciphertext		AES-CBC: (A4501, A4493, A4484, A4489, A4497, A4487) AES-CTR: (A4501, A4493, A4484,

Name	Type	Description	Properties	Algorithms
				A4489, A4497, A4487) AES-ECB: (A4500, A4501, A4495, A4494, A4496, A4504, A4485, A4502, A4486, A4498, A4488, A4499, A4493, A4484, A4489, A4490, A4503, A4497, A4487) AES-XTS Testing Revision 2.0: (A4501, A4493, A4484, A4489, A4497, A4487)
Random Number Generation	DRBG	Random Number Generation		Counter DRBG: (A4501, A4493, A4484, A4489, A4497, A4487)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

GCM IV

The module offers three AES GCM implementations. The GCM IV generation for these implementations complies respectively with IG C.H under Scenario 1, Scenario 2, and Scenario 5. The GCM shall only be used in the context of the AES-GCM encryption executing under each scenario.

Scenario 1, TLS 1.2

For TLS 1.2, the module offers the GCM implementation and uses the context of Scenario 1 of IG C.H. The module is compliant with SP800-52rev2 and the mechanism for IV generation is compliant with RFC5288. The module supports acceptable AES-GCM ciphersuites from Section 3.3.1 of SP800-52rev2.

The module explicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values of $2^{\{64-1\}}$ for a given session key. If this 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.

Scenario 2, Random IV

In this implementation, the module offers the interfaces `EVP_aead_aes_128_gcm_randnonce()` and `EVP_aead_aes_256_gcm_randnonce()` for compliance with Scenario 2 of IG C.H and SP800-38D Section 8.2.2. The 96-bit AES-GCM IV,

containing 96 bits of entropy, is generated randomly internal to the module using module's approved DRBG, without outputting the IV to the calling application. The DRBG is seeded from the entropy source.

Scenario 5, TLS 1.3

August 2018, using the ciphersuites that explicitly select AES-GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES-GCM ciphersuites from Section 3.3.1 of SP800-52rev2.

The module implements, within its boundary, an IV generation unit for TLS 1.3 that keeps control of the 64-bit counter value within the AES-GCM IV. If the exhaustion condition is observed, the module will return an error indication to the calling application, who will then need to either trigger a re-key of the session (i.e., a new key for AES-GCM), or terminate the connection.

In the event the module's power is lost and restored, the consuming application must ensure that new AES-GCM keys encryption or decryption under this scenario are established. TLS 1.3 provides session resumption, but the resumption procedure derives new AES-GCM encryption keys.

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. 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. To meet the requirement stated in IG C.I, the module implements a check to ensure 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.

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 with SP 800-132 and FIPS 140-3 IG D.N, the following requirements shall be met:

- Derived keys shall only be used in storage applications. The MK shall not be used for other purposes. The module accepts a minimum length of 112 bits for the MK or DPK.
- Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic Keys.
- The minimum length of the password or passphrase accepted by the module is 14 characters. This results in the estimated probability of guessing the password to be at most 10^{-14} . Combined with the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.
- A portion of the salt, with a length of at least 128 bits (this is verified by the module to determine the service is approved), shall be generated randomly using the SP 800-90Ar1 DRBG provided by the module.
- The iteration count shall be selected as large as possible, if the time required to generate the key using the entered password is acceptable for the users. The module restricts the minimum iteration count to be 1000.

Compliance to SP 800-56Arev3 assurances

The module offers ECDH shared secret computation services compliant to the SP 800-56Arev3 and meeting IG D.F scenario 2 path (1). To meet the required assurances listed in section 5.6 of SP 800-56Arev3, the module shall be used together with an application that implements the "TLS protocol" and the following steps shall be performed.

- The entity using the module, must use the module's "Key Pair Generation" service for generating ECDH ephemeral keys. This meets the assurances required by key pair owner defined in the section 5.6.2.1 of SP 800-56Arev3.
- As part of the module's shared secret computation (SSC) service, the module internally performs the public key validation on the peer's public key passed in as input to the SSC function. This meets the public key validity assurance required by the sections 5.6.2.2.1/5.6.2.2.2 of SP 800-56Arev3.
- The module does not support static keys therefore the "assurance of peer's possession of private key" is not applicable.

Key Establishment

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

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.

Legacy Algorithms

According to IG C.M, digital signature verification using RSA with a 1024-bits modulus is approved for legacy usage only. Digital signature verification using SHA-1 is approved for legacy usage only. The CAVP certificates for these algorithms are listed in the *Approved Algorithms* table. These legacy algorithms can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

SHA-1 is only approved in the module when used for Message Digest, Message Authentication Generation with HMAC, HKDF Key Derivation, SSH Key Derivation and Signature Verification (Legacy). The use of SHA-1 for Digital Signature Generation is non-approved in the module.

See section 9.5 regarding algorithm transitions.

Approved Modulus Sizes for RSA Digital Signature

RSA SigGen (FIPS 186-5) has been CAVP tested with all the supported RSA modulus lengths (i.e., 2048, 3072, 4096). This is documented in the Approved Algorithms table of the Security Policy. All modulus sizes for SigVer have also been CAVP tested. There is no RSA signature with keys for which CAVP testing is not available. The minimum number of the Miller-Rabin tests used in the primality testing complies with Table B.1 in FIPS 186-5. The RSA SigVer (186-4) and (FIPS 186-5) have been CAVP tested with the modulus sizes of 1024 (FIPS 186-4) and 2048, 3072, 4096 (FIPS 186-5). All modulus sizes in which testing is available have been tested by the CAVP.

2.8 RBG and Entropy

N/A for this module.

N/A for this module.

The module provides an SP800-90Arev1-compliant Deterministic Random Bit Generator (DRBG) using CTR_DRBG mechanism with AES-256 for generation of key components of asymmetric keys, and random number generation. The DRBG is seeded with 256-bit of entropy input provided from an external entity to the module. This corresponds to scenario 2 (b) of IG 9.3.A i.e., the DRBG that receives a LOAD command with entropy obtained from inside the physical perimeter of the operational environment but outside of module's cryptographic boundary. The calling application shall use an entropy source that meets the security strength required for the CTR_DRBG as shown in NIST SP 800-90Arev1, Table 3 and should return an error if minimum strength cannot be met.

Per the IG 9.3.A requirement, the module includes the caveat "No assurance of the minimum strength of generated keys".

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133Rev2. When random values are required, they are obtained from the SP 800-90Arev1 approved DRBG, compliant with Section 4 of SP 800-133Rev2, as documented in the Vendor-Affirmed Algorithms Table.

Additionally, the module implements methods for key derivation and key generation as documented in the Security Function Implementations Table.

2.10 Key Establishment

The module implements methods that may be used as part of a Key Agreement Scheme or a Key Transport Scheme as listed 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.

No parts of the SSH, TLS, 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 for data.
N/A	Data Output	API output parameters for data.
N/A	Control Input	API function calls.
N/A	Status Output	API return codes, error message.

Table 10: Ports and Interfaces

As a Software module, the module interfaces are defined as Software or Firmware Module Interfaces (SMFI), and there are no physical ports. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not support authentication.

4.2 Roles

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

Table 11: Roles

The module does not support concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Encryption	Encryption	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key, plaintext	Ciphertext	Symmetric Encryption	Crypto Officer - AES Key: W,E
Decryption	Decryption	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key, ciphertext	Plaintext	Symmetric Decryption	Crypto Officer - AES Key: W,E
Authenticated Encryption	Authenticated encryption	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key, IV, plaintext	Ciphertext, MAC tag	Authenticated Encryption	Crypto Officer - AES Key: W,E
Authenticated Decryption	Authenticated decryption	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key, ciphertext, MAC tag, IV	Plaintext	Authenticated Decryption	Crypto Officer - AES Key: W,E
Message Authentication Generation	MAC generation	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key/HMAC key, plaintext	MAC tag	Message Authentication Generation with AES Message Authentication Generation with HMAC	Crypto Officer - HMAC Key: W,E - AES Key: W,E
Message Authentication Verification	MAC verification	Return value 1 from the function: FIPS_service_indicator_check_approved()	AES key, ciphertext, MAC tag, IV	Plaintext	Message Authentication Verification with AES-GMAC	Crypto Officer - AES Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Digest	Generating message digest	Return value 1 from the function: FIPS_service_indicator_check_approved()	Message	Message digest	Message Digest	Crypto Officer
Random Number Generation	Generating random numbers	Return value 1 from the function: FIPS_service_indicator_check_approved()	Output length	Random bytes	Random Number Generation	Crypto Officer - Entropy Input (per IG D.L): W,E - DRBG Seed (per IG D.L): G,E - DRBG Internal State (V, Key) (per IG D.L): G,E
Key Generation	Generating key pair	Return value 1 from the function: FIPS_service_indicator_check_approved()	RSA: Modulus size; ECDSA: Curve	RSA: Module generated RSA public key, Module generated RSA private key; ECDSA: Module Generated EC Private Key, Module Generated EC Public Key	ECDSA Key Generation RSA Key Generation	Crypto Officer - Module generated RSA Public Key: G,R - Module generated RSA Private Key: G,R - Module generated EC Public Key: G,R - Module generated EC Private Key: G,R - Intermediate Key Generation Value: G,E,Z
Key Verification	Verifying the public key	Return value 1 from the function: FIPS_service_indicator_check_approved()	Public key	Success/error	ECDSA Key Verification	Crypto Officer - EC Public Key: W,E
Signature Generation	Generating signature	Return value 1 from the function:	Message, EC private	Digital signature	ECDSA Signature Generation	Crypto Officer - EC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		FIPS_service_indicator_check_approved()	key or RSA private key		RSA Signature Generation	Private Key: W,E - RSA Private Key: W,E
Signature Verification	Verifying signature	Return value 1 from the function: FIPS_service_indicator_check_approved()	Signature, EC public key or RSA public key	Digital signature verification result	ECDSA Signature Verification Legacy ECDSA Signature Verification Legacy RSA Signature Verification RSA Signature Verification	Crypto Officer - EC Public Key: W,E - RSA Public Key: W,E
Shared Secret Computation	Calculating the Shared Secret	Return value 1 from the function: FIPS_service_indicator_check_approved()	EC public key, EC private key	Shared Secret	KAS-ECC-SSC (Shared Secret Computation with EC Diffie-Hellman)	Crypto Officer - EC Public Key: W,E - EC Private Key: W,E - Shared Secret: G,R
TLS KDF Key Derivation	Deriving keys	Return value 1 from the function: FIPS_service_indicator_check_approved()	TLS Pre-Master Secret, TLS Master Secret	TLS Master Secret, TLS Derived Key	TLS Key Derivation	Crypto Officer - TLS Pre-Master Secret: W,E - TLS Master Secret : G,E,R,Z - TLS Derived Key: G,R,Z
HKDF Key Derivation	Deriving keys	Return value 1 from the function: FIPS_service_indicator_check_approved()	Shared Secret	HKDF Derived key	HKDF Key Derivation	Crypto Officer - Shared Secret: W,E - KDA HKDF derived key: G,R
SSH Key Derivation	Deriving keys	Return value 1 from the function:	Shared Secret	SSH KDF Derived Key	SSH Key Derivation	Crypto Officer - Shared

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
		FIPS_service_indicator_check_approved()				Secret: W,E - SSH KDF Derived Key: G,R
PBKDF Key Derivation	Deriving keys	Return value 1 from the function: FIPS_service_indicator_check_approved()	Password	PBKDF Derived Key	PBKDF Key Derivation	Crypto Officer - Password: W,E - PBKDF Derived Key: G,R
Zeroization	Zeroize PSP in volatile memory	N/A	SSP	N/A	None	Crypto Officer - AES Key: Z - HMAC Key: Z - Entropy Input (per IG D.L): Z - DRBG Seed (per IG D.L): Z - DRBG Internal State (V, Key) (per IG D.L): Z - RSA Public Key: Z - RSA Private Key: Z - Module generated RSA Public Key: Z - Module generated RSA Private Key: Z - EC Public Key: Z - EC Private Key: Z - Module generated EC Public Key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Module generated EC Private Key: Z - Shared Secret: Z - TLS Pre-Master Secret: Z - TLS Master Secret : Z - TLS Derived Key: Z - KDA HKDF derived key: Z - Password: Z - Intermediate Key Generation Value: Z - SSH KDF Derived Key: Z - PBKDF Derived Key: Z
On-Demand Self-test	Initiate power-on self-tests by reset	N/A	N/A	Pass or fail	ECDSA Key Generation RSA Key Generation ECDSA Key Verification ECDSA Signature Generation RSA Signature Generation ECDSA Signature Verification Legacy ECDSA Signature Verification Legacy RSA Signature	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Verification RSA Signature Verification Message Authentic ation Generation with AES Message Authentic ation Generation with HMAC KAS-ECC- SSC (Shared Secret Computati on with EC Diffie- Hellman) HKDF Key Derivation SSH Key Derivation TLS Key Derivation PBKDF Key Derivation Message Digest Authentic ated Encryption Authentic ated Decryption Symmetric Encryption Symmetric Decryption Random Number Generation	
On-Demand Integrity Test	Initiate integrity test on-demand	N/A	N/A	Pass or fail	None	Crypto Officer
Show Status	Show status of the module state	N/A	N/A	Module status	None	Crypto Officer
Show Version	Show the version of the module using awslc_version_string	N/A	N/A	Module name and version	None	Crypto Officer

Table 12: Approved Services

For the above table, the convention below applies when specifying the access permissions (types) that the service has for each SSP.

- R = Read: The SSP is read from the module (e.g., the SSP is output).
- W = Write: The SSP is updated, imported, or written to the module.
- E = Execute: The module uses the SSP in performing a cryptographic operation.
- Z = Zeroize: The module zeroizes the SSP.

For the role, CO indicates “Crypto Officer”.

The module implements a service indicator that indicates whether the invoked service is approved. The service indicator is a return value 1 from the `FIPS_service_indicator_check_approved` function. This function is used together with two other functions. The usage is as follows:

- STEP 1: Should be called before invoking the service.
`int before = FIPS_service_indicator_before_call();`
- STEP 2: Make a service call i.e., API function for performing a service.
`Func();`
- STEP 3: Should be called after invoking the service.
`int after = FIPS_service_indicator_after_call();`
- STEP 4: Return value 1 indicates approved service was invoked.
`int ret = FIPS_service_indicator_check_approved(before, after);`

Alternatively, all the above steps can be done by using a single call using the function `CALL_SERVICE_AND_CHECK_APPROVED(approved, func)`.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES with OFB or CFB1, CFB8 modes	Encryption, Decryption	AES with OFB or CFB1, CFB8 modes	CO
AES-GCM, CCM, GMAC, XTS keys not listed in the Approved Services table of the Security Policy	Encryption, Decryption	AES-GCM, CCM, GMAC, XTS keys not listed in the Approved Services table of the Security Policy	CO
AES using <code>aes_*_generic</code> function	Encryption, Decryption	AES using <code>aes_*_generic</code> function	CO
AES GMAC using <code>aes_*_generic</code>	Message Authentication Generation	AES GMAC using <code>aes_*_generic</code>	CO
Curve <code>secp256k1</code>	Signature Generation, Signature Verification, Shared Secret Computation	Curve <code>secp256k1</code>	CO
Diffie Hellman	Shared Secret Computation	Diffie Hellman	CO
HMAC-MD4, HMAC-MD5, HMAC-SHA1, HMAC-SHA-3, HMAC-RIPEMD-160	Message Authentication Generation	HMAC-MD4, HMAC-MD5, HMAC-SHA1, HMAC-SHA-3, HMAC-RIPEMD-160	CO

Name	Description	Algorithms	Role
MD4	Message Digest	MD4	CO
MD5	Message Digest (outside of TLS)	MD5	CO
RSA using RSA_generate_key_ex	Key Generation	RSA using RSA_generate_key_ex	CO
ECDSA using EC_KEY_generate_key	Key Generation	ECDSA using EC_KEY_generate_key	CO
RSA using keys less than 2048 bits	Signature Generation	RSA using keys less than 2048 bits	CO
RSA using keys less than 1024 bits	Signature Verification	RSA using keys less than 1024 bits	CO
RSA	Key Encapsulation/Un-encapsulation, sign/verify primitive operations without hashing	RSA	CO
RSA with PKCS#1 v1.5 and OAEP padding	Encryption primitive	RSA with PKCS#1 v1.5 and OAEP padding	CO
SHA-1, SHA-3	Signature Generation	SHA-1, SHA-3	CO
SHAKE, RIPEMD-160, SHA-3	Message Digest	SHAKE, RIPEMD-160, SHA-3	CO
TLS KDF using any SHA algorithms not listed in SP the Approved Services table of the Security Policy or TLS KDF using non extended master secret	Key Derivation	TLS KDF using any SHA algorithms not listed in SP the Approved Services table of the Security Policy or TLS KDF using non extended master secret	CO

Table 13: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not support loading of external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC value calculated at run time on the bcm.o file, with the HMAC-SHA2-256 value stored within the module that was computed at build time.

5.2 Initiate on Demand

The module provides on-demand integrity test. The integrity test can be performed on demand by reloading the module. Additionally, the integrity test can be performed using the On-Demand Integrity Test service, which calls the BORINGSSL_integrity_test function.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

The module must be compiled and installed as stated in section 11. The user should confirm that the module is installed correctly by following steps 4 and 5 listed in section 11.

6.2 Configuration Settings and Restrictions

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

The module claims no non-invasive security techniques.

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. The module does not perform persistent storage of SSPs	Dynamic

Table 14: Storage Areas

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 15: SSP Input-Output Methods

The module does not support entry and output of SSPs beyond the physical perimeter of the operational environment. The SSPs are provided to the module via API input parameters in the plaintext form and output via API output parameters in the plaintext form to and from the calling application running on the same operational environment.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Wipe and Free memory block allocated	Zeroizes the SSPs contained within the cipher handle.	Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization procedure succeeded.	By calling the cipher related zeroization API
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 16: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	AES Key	128-256 bits - 128-256 bits	Symmetric key - CSP			Message Authentication Generation with AES Message Authentication Verification with AES-GMAC Authenticated Encryption Symmetric Encryption Authenticated Decryption Symmetric Decryption
HMAC Key	HMAC Key	112-256 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Generation with HMAC
Entropy Input (per IG D.L)	Entropy Input (per IG D.L)	256 bits - 256 bits	Entropy Input - CSP			Random Number Generation
DRBG Seed (per IG D.L)	DRBG Seed (per IG D.L)	256 bits - 256 bits	Seed - CSP	Random Number Generation		Random Number Generation
DRBG Internal State (V, Key) (per IG D.L)	DRBG Internal State (V, Key) (per IG D.L)	256 bits - 256 bits	Internal state - CSP	Random Number Generation		Random Number Generation
RSA Public Key	RSA Public Key	1024, 2048, 3072, 4096 bits - 80-150 bits	Public key - PSP			Legacy RSA Signature Verification RSA Signature Verification
RSA Private Key	RSA Private Key	2048, 3072, 4096 bits - 112-150 bits of strength	Private key - CSP			RSA Signature Generation
Module generated RSA Public Key	Module generated RSA Public Key	2048, 3072, 4096 bits - 112-150 bits	Public key - PSP	RSA Key Generation		
Module generated RSA Private Key	Module generated RSA Private Key	2048, 3072, 4096 bits - 112-150 bits	Private key - CSP	RSA Key Generation		
EC Public Key	EC Public Key	P-224, P-256, P-384, P-521 - 112-256 bits	Public key - PSP			ECDSA Key Verification ECDSA Signature Verification Legacy ECDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Signature Verification KAS-ECC-SSC (Shared Secret Computation with EC Diffie-Hellman)
EC Private Key	EC Private Key	P-224, P-256, P-384, P-521 - 112-256 bits	Private key - CSP			ECDSA Signature Generation KAS-ECC-SSC (Shared Secret Computation with EC Diffie-Hellman)
Module generated EC Public Key	Module generated EC Public Key	P-224, P-256, P-384, P-521 - 112-256 bits	Public key - PSP	ECDSA Key Generation		
Module generated EC Private Key	Module generated EC Private Key	P-224, P-256, P-384, P-521 - 112-256 bits	Private key - CSP	ECDSA Key Generation		
Shared Secret	Shared Secret	P-224, P-256, P-384, P-521 - 112-256 bits	Shared Secret - CSP	KAS-ECC-SSC (Shared Secret Computation with EC Diffie-Hellman)		HKDF Key Derivation SSH Key Derivation TLS Key Derivation PBKDF Key Derivation
TLS Pre-Master Secret	TLS Pre-Master Secret	P-224, P-256, P-384, P-521 - 112-256 bits	TLS pre-master secret - CSP			TLS Key Derivation
TLS Master Secret	TLS Master Secret	384 bits - 112-256 bits	TLS master secret - CSP	TLS Key Derivation		HKDF Key Derivation SSH Key Derivation TLS Key Derivation PBKDF Key Derivation
TLS Derived Key	Generated by TLS KDF key derivation	112-256 bits - 112-256 bits	Symmetric key - CSP	TLS Key Derivation		
KDA HKDF derived key	KDA HKDF derived key	112-2048 bits - 112-256 bits	Symmetric key - CSP	HKDF Key Derivation		
SSH KDF Derived Key	SSH KDF Derived Key	112-256 bits - 112-256 bits	Symmetric key - CSP	SSH Key Derivation		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
PBKDF Derived Key	PBKDF Derived Key	112-4096 bits - 112-256 bits of strength	Symmetric key - CSP	PBKDF Key Derivation		
Password	Password	112-1024 bits - N/A	Password - CSP			PBKDF Key Derivation
Intermediate Key Generation Value	Intermediate Key Generation Value	224-4096 bits - 112-256 bits	Intermediate value - CSP	ECDSA Key Generation RSA Key Generation		ECDSA Key Generation RSA Key Generation

Table 17: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
HMAC Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	
Entropy Input (per IG D.L)	API input parameters	RAM:Plaintext	From service invocation to service completion	Automatic	DRBG Seed (per IG D.L):Generates
DRBG Seed (per IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic	Entropy Input (per IG D.L):Generated From DRBG Internal State (V, Key) (per IG D.L):Generates
DRBG Internal State (V, Key) (per IG D.L)		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	DRBG Seed (per IG D.L):Generated From
RSA Public Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA Private Key:Paired With
RSA Private Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	RSA Public Key:Paired With
Module generated RSA Public Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Module generated RSA Private Key:Paired With Intermediate Key Generation Value:Generated From
Module generated RSA Private Key	API output parameters	RAM:Plaintext	From service invocation to	Wipe and Free memory block	Module generated RSA Public Key:Paired With Intermediate Key

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			service completion	allocated Module Reset	Generation Value:Generated From
EC Public Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC Private Key:Paired With Shared Secret:Generates
EC Private Key	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC Public Key:Paired With Shared Secret:Generates
Module generated EC Public Key	API output parameters	RAM:Plaintext	From service invocation to service completion		Module generated EC Private Key:Paired With Intermediate Key Generation Value:Generated From Shared Secret:Generates
Module generated EC Private Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Module generated EC Public Key:Paired With Intermediate Key Generation Value:Generated From Shared Secret:Generates
Shared Secret	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	EC Public Key:Derived From EC Private Key:Derived From Shared Secret:Derived From
TLS Pre-Master Secret	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	TLS Master Secret :Derives
TLS Master Secret		RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	TLS Pre-Master Secret:Derived From TLS Derived Key:Derives
TLS Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	TLS Master Secret :Derived From
KDA HKDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared Secret:Derived From
SSH KDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	Shared Secret:Derived From
PBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to	Wipe and Free memory block	Password:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			service completion	allocated Module Reset	
Password	API input parameters	RAM:Plaintext	From service invocation to service completion	Wipe and Free memory block allocated Module Reset	PBKDF Derived Key:Derives
Intermediate Key Generation Value		RAM:Plaintext	From service invocation to service completion	Automatic	Module generated RSA Public Key:Generates Module generated RSA Private Key:Generates Module generated EC Public Key:Generates Module generated EC Private Key:Generates

Table 18: 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.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A4483)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4491)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4492)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4505)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4506)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A
HMAC-SHA2-256 (A4507)	SHA2-256	Message Authentication	SW/FW Integrity	Module becomes operational	N/A

Table 19: Pre-Operational Self-Tests

The module performs the pre-operational self-test automatically when the module is loaded into memory; the pre-operational self-test is the software integrity test that ensures that the module is not corrupted. While the module is executing the pre-operational self-test, services are not available, and input and output are inhibited.

The software integrity test is performed after a set of conditional cryptographic algorithm self-tests (CASTs). The set of CASTs executed before the software integrity test consists of HMAC-SHA2-256 KAT, which is used in the pre-operational self-test, and the SHA2-256 KAT.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A4501) - Encryption	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4493) - Encryption	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4484) - Encryption	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4489) - Encryption	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4497) - Encryption	128-bit keys, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4487) - Encryption	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 (A4500) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4495) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4494) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4496) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4504) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4485) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4502) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4486) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4498) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4488) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4499) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4490) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4503) - Encryption	128-bit keys, 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4501) - Decryption	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4493) - Decryption	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4484) - Decryption	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4489) - Decryption	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-CBC (A4497) - Decryption	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-CBC (A4487) - Decryption	128-bit keys, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4500) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4495) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4494) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4496) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4504) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4485) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4502) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4486) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4498) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4488) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4499) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4490) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4503) - Decryption	128-bit keys, 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
SHA-1 (A4483)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4492)	SHA-1	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
SHA-1 (A4505)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4491)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4508)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4506)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4507)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4483)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4492)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4505)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4491)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4508)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4506)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A4507)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4483)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4492)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4505)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4491)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4508)	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 (A4506)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4507)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4483)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4492)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4505)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4491)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4508)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4506)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A4507)	SHA-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
Counter DRBG (A4501)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A4493)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A4484)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A4489)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A4497)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Counter DRBG (A4487)	AES-256 without prediction resistance	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4483)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4492)	SHA-256; P-256	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
ECDSA SigGen (FIPS186-5) (A4505)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4491)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4508)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4506)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A4507)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4483)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4492)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4505)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4491)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4508)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4506)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A4507)	SHA-256; P-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4483)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4492)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4505)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4491)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4508)	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 (A4506)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4507)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-5) (A4483)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4492)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4505)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4491)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4508)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4506)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A4507)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
KDF TLS (A4483)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4492)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4505)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4491)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4508)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4506)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDF TLS (A4507)	SHA2-256	KAT	CAST	Module becomes operational	Key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4483)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret 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 (A4492)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4505)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4491)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4508)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4506)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4507)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
PBKDF (A4483)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4492)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4505)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4491)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4508)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4506)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A4507)	SHA2-256	KAT	CAST	Module becomes operational	Password-based key derivation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4483)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4492)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4505)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4491)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	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) (A4508)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4506)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A4507)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4483)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4492)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4505)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4491)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4508)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4506)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A4507)	PKCS#1 v1.5 with 4096 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA KeyGen (FIPS186-5) (A4483)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4492)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4505)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4491)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4508)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4506)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A4507)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation

Table 20: Conditional Self-Tests

Conditional Cryptographic Algorithm Tests

The module performs self-tests on approved cryptographic algorithms, using the tests shown in Table 22. Data output through the data output interface is inhibited during the self-tests. The CASTs are performed in the form of Known Answer Tests (KATs), in which the calculated output is compared with the expected known answer (that are hard-coded in the module). A failed match causes a failure of the self-test. If any of these self-tests fails, the module transitions to error state.

Conditional Pair-Wise Consistency Tests

The module implements RSA and ECDSA key generation service and performs the respective pairwise consistency test (PCT) using sign and verify functions when the keys are generated (Table 22). If any of these self-tests fails, the module transitions to error state and is aborted.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4483)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4491)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4492)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4505)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4506)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A4507)	Message Authentication	SW/FW Integrity	On demand	Manually

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A4501) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4493) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4484) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4489) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4497) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4487) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4500) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4495) - Encryption	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A4494) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4496) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4504) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4485) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4502) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4486) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4498) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4488) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4499) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4490) - Encryption	KAT	CAST	On Demand	Manually
AES-GCM (A4503) - Encryption	KAT	CAST	On Demand	Manually
AES-CBC (A4501) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A4493) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A4484) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A4489) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A4497) - Decryption	KAT	CAST	On Demand	Manually
AES-CBC (A4487) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4500) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4495) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4494) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4496) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4504) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4485) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4502) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4486) - Decryption	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A4498) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4488) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4499) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4490) - Decryption	KAT	CAST	On Demand	Manually
AES-GCM (A4503) - Decryption	KAT	CAST	On Demand	Manually
SHA-1 (A4483)	KAT	CAST	On Demand	Manually
SHA-1 (A4492)	KAT	CAST	On Demand	Manually
SHA-1 (A4505)	KAT	CAST	On Demand	Manually
SHA-1 (A4491)	KAT	CAST	On Demand	Manually
SHA-1 (A4508)	KAT	CAST	On Demand	Manually
SHA-1 (A4506)	KAT	CAST	On Demand	Manually
SHA-1 (A4507)	KAT	CAST	On Demand	Manually
SHA2-256 (A4483)	KAT	CAST	On Demand	Manually
SHA2-256 (A4492)	KAT	CAST	On Demand	Manually
SHA2-256 (A4505)	KAT	CAST	On Demand	Manually
SHA2-256 (A4491)	KAT	CAST	On Demand	Manually
SHA2-256 (A4508)	KAT	CAST	On Demand	Manually
SHA2-256 (A4506)	KAT	CAST	On Demand	Manually
SHA2-256 (A4507)	KAT	CAST	On Demand	Manually
SHA2-512 (A4483)	KAT	CAST	On Demand	Manually
SHA2-512 (A4492)	KAT	CAST	On Demand	Manually
SHA2-512 (A4505)	KAT	CAST	On Demand	Manually
SHA2-512 (A4491)	KAT	CAST	On Demand	Manually
SHA2-512 (A4508)	KAT	CAST	On Demand	Manually
SHA2-512 (A4506)	KAT	CAST	On Demand	Manually
SHA2-512 (A4507)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4483)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4492)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4505)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4491)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4508)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4506)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A4507)	KAT	CAST	On Demand	Manually
Counter DRBG (A4501)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG (A4493)	KAT	CAST	On Demand	Manually
Counter DRBG (A4484)	KAT	CAST	On Demand	Manually
Counter DRBG (A4489)	KAT	CAST	On Demand	Manually
Counter DRBG (A4497)	KAT	CAST	On Demand	Manually
Counter DRBG (A4487)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4483)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4492)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4505)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4491)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4508)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4506)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A4507)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4483)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4492)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4505)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4491)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4508)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4506)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A4507)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KAS-ECC-SSC Sp800-56Ar3 (A4483)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4492)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4505)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4491)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4508)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4506)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4507)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4483)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4492)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4505)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4491)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4508)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4506)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A4507)	PCT	PCT	On Demand	Manually
KDF TLS (A4483)	KAT	CAST	On Demand	Manually
KDF TLS (A4492)	KAT	CAST	On Demand	Manually
KDF TLS (A4505)	KAT	CAST	On Demand	Manually
KDF TLS (A4491)	KAT	CAST	On Demand	Manually
KDF TLS (A4508)	KAT	CAST	On Demand	Manually
KDF TLS (A4506)	KAT	CAST	On Demand	Manually
KDF TLS (A4507)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800- 56Cr1 (A4483)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDA HKDF Sp800-56Cr1 (A4492)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4505)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4491)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4508)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4506)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4507)	KAT	CAST	On Demand	Manually
PBKDF (A4483)	KAT	CAST	On Demand	Manually
PBKDF (A4492)	KAT	CAST	On Demand	Manually
PBKDF (A4505)	KAT	CAST	On Demand	Manually
PBKDF (A4491)	KAT	CAST	On Demand	Manually
PBKDF (A4508)	KAT	CAST	On Demand	Manually
PBKDF (A4506)	KAT	CAST	On Demand	Manually
PBKDF (A4507)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4483)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4492)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4505)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4491)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4508)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4506)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A4507)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4483)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4492)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4505)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) (A4491)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4508)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4506)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A4507)	KAT	CAST	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4483)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4492)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4505)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4491)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4508)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4506)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A4507)	PCT	PCT	On Demand	Manually

Table 22: Conditional Periodic Information

The module does not support periodic self-tests.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The library is aborted with SIGABRT signal. Module is no longer operational the data output interface is inhibited	Pre-operational test failure Conditional test failure	Module reset	Per-operational test failure: Error message is output (i.e., "FIPS integrity test failed.") on the stderr and then the module is aborted; Conditional test failure: For CAST failure, an error message (i.e., "** KAT failed") is output on the stderr and then the module is aborted. For PCT failure, an error (i.e., "EC_R_PUBLIC_KEY_VALIDATION_FAILED" or "RSA_R_PUBLIC_KEY_VALIDATION_FAILED") is output in the error queue and then the module generates new key, If the PCT still does not pass, eventually the

Name	Description	Conditions	Recovery Method	Indicator
				module will be aborted after 5 tries. The abort statement indicates that the module has been aborted.

Table 23: Error States

If the module fails any of the self-tests, the module enters an error state. To recover from any error state, the module must be rebooted.

10.5 Operator Initiation of Self-Tests

The software integrity tests and the CASTs for AES, SHS, DRBG, HMAC, KAS-ECC-SSC, TLS KDF, KDA HKDF, PBKDF2 can be invoked by unloading and subsequently re-initializing the module. The CASTs for ECDSA and RSA can be invoked by requesting the corresponding Key Generation or Digital Signature services. Additionally, all the CASTs can be invoked by calling the BORINGSSL_self_test function. The PCTs can be invoked on demand by requesting the Key Generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module `bcm.o` is embedded into the shared library `libcrypto.so` which can be obtained by building the source code at the following location [1]. The set of files specified in the archive constitutes the complete set of source files of the validated module. There shall be no additions, deletions, or alterations of this set as used during module build.

[1] <https://github.com/aws/aws-lc/archive/refs/tags/AWS-LC-FIPS-2.0.0.zip>

The downloaded zip file can be verified by issuing the “`sha256sum AWS-LC-FIPS-2.0.0.zip`” command. The expected SHA2-256 digest value is:

6241EC2F13A5F80224EE9CD8592ED66A97D426481066FEAA4EFC6F24E60BBC96

After the zip file is extracted, the instructions listed below will compile the module. The compilation instructions must be executed separately on platforms that have different processors and/or operating systems. Due to six possible combinations of OS/processor, the module count is six (i.e., there are six separate binaries generated, one for each entry listed in the Tested Operational Environments table).

Amazon Linux 2 and Amazon Linux 2023:

1. `sudo yum groupinstall "Development Tools"`
2. `sudo yum install cmake3 golang`
3. `cd aws-lc-fips-2022-11-02/`
4. `mkdir build`
5. `cd build`
6. `cmake3 -DFIPS=1 ..`
7. `make`

Ubuntu 22.04:

1. `sudo apt-get install build-essential`
2. `sudo apt-get install cmake`
3. Get latest Golang archive for your architecture
4. `sudo tar -C /usr/local -xzf go*.tar.gz`
5. `cd aws-lc-fips-2022-11-02/`
6. `mkdir build`
7. `cd build`
8. `cmake -DFIPS=1 -DGO_EXECUTABLE=/usr/local/go/bin/go ..`
9. `make`

Upon completion of the build process, the module’s status can be verified by the command below. If the value obtained is “1” then the module has been installed and configured to operate in FIPS compliant manner.

`./tool/bssl isfips`

Lastly, the user can call the “show version” service using `awslc_version_string` function and the expected output is “AWS-LC FIPS 2.0.0” which is the module version. This will confirm that the module is in the operational mode. Additionally, the “AWS-LC FIPS” also acts as the module identifier and the verification of the “dynamic” part can be done using following command with an application that was used for dynamic linking. The “U” in the output confirms that the module is dynamically linked.

Command: `nm <application_name> | grep awslc_version_string`

Example Output: “`U awslc_version_string`”

11.2 Administrator Guidance

When the module is at end of life, for the GitHub repo, the README will be modified to mark the library as deprecated. After a 6-month window, more restrictive branch permissions will be added such that only administrators can read from the FIPS branch.

The module does not possess persistent storage of SSPs. The SSP value only exists in volatile memory and that value vanishes when the module is powered off. So as a first step for the secure sanitization, the module needs to be powered off. Then for actual deprecation, the module will be upgraded to newer version that is approved. This upgrade process will uninstall/remove the old/terminated module and provide a new replacement.

The Approved and non-Approved modes of operation are specified in section 2.4. The administrative functions are specified in the Approved Services table. All the logical interfaces are specified in section 3.1. The requirements and restrictions that shall be considered when operating the module in approved mode are specified in section 2.7 and section 6. The installation, initialization, and startup procedures specified in section 11.1 shall be followed.

12 Mitigation of Other Attacks

12.1 Attack List

RSA timing attacks.

12.2 Mitigation Effectiveness

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

The module provides the mechanism to use the blinding for RSA. When the blinding is on, the module generates a random value to form a blinding factor in the RSA key before the RSA key is used in the RSA cryptographic operations.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
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
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
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
IPsec	Internet Protocol Security
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
KW	Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS 140-3	FIPS PUB 140-3 - Security Requirements For Cryptographic Modules March 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG (Last Update: September 2, 2025)	Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	Secure Hash Standard (SHS) March 2012 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS 186-4	Digital Signature Standard (DSS) July 2013 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS 186-5	Digital Signature Standard (DSS) February 2023 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf
FIPS 197	Advanced Encryption Standard May 9, 2023 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
FIPS 198-1	The Keyed Hash Message Authentication Code (HMAC) July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 February 2003 https://www.ietf.org/rfc/rfc3447.txt
RFC 3526	More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE) May 2003 https://www.ietf.org/rfc/rfc3526.txt
RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) June 2005 https://datatracker.ietf.org/doc/html/rfc4106
RFC 7296	Internet Key Exchange Protocol Version 2 (IKEv2) October 2014 https://datatracker.ietf.org/doc/html/rfc7296
SP 800-38A	Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001 https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP 800-38A Addendum	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode October 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf

SP 800-38B	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication May 2005 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38B.pdf
SP 800-38C	Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality May 2004 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP 800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC November 2007 https://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP 800-38E	Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices January 2010 https://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP 800-38F	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping December 2012 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP 800-56Ar3	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography April 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
SP 800-90Ar1	Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP 800-90B	Recommendation for the Entropy Sources Used for Random Bit Generation January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP 800-133r2	Recommendation for Cryptographic Key Generation June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP 800-140Br1	CMVP Security Policy Requirements March 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf