AWS-LC Cryptographic Module (dynamic)

Module Version: AWS-LC FIPS 2.0.0

FIPS 140-3 Non-Proprietary Security Policy

Document version: 1.0

Last update: 2024-08-13

Prepared by:
atsec information security corporation
4516 Seton Center Pkwy, Suite 250
Austin, TX 78759
www.atsec.com

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

1 Table of Contents

1 General 7
1.1 Overview 7
1.2 Security Levels 7
1.3 Additional Information 7
2 Cryptographic Module Specification 9
2.1 Description 9
2.2 Operating Environments 9
2.3 Excluded Components 10
2.4 Modes of Operation of the Module 10
2.5 Algorithms 11
2.6 Security Function Implementation 16
2.7 Algorithm Specific Information 17
2.7.1 GCM IV 17
2.7.2 AES XTS 18
2.7.3 Key Derivation using SP 800-132 PBKDF2 18
2.7.4 Compliance to SP 800-56ARev3 assurances 18
2.7.5 Approved Modulus Sizes for RSA Digital Signature 19
2.7.6 Legacy Algorithms 19
2.8 RNG and Entropy 19
2.9 Key Generation 19
2.10 Key Establishment 20
2.11 Industry Protocols 20
3 Cryptographic Module Interfaces 21
4 Roles, Services, and Authentication 22
4.1 Authentication Methods 22
4.2 Roles 22
4.3 Approved Services 22
4.4 Non-Approved Services 24
4.5 External Software/Firmware Loaded 25
5 Software/Firmware Security 26
5.1 Integrity Techniques 26
5.2 Initiate On-Demand Integrity Test 26
6 Operational Environment 27
6.1 Operational Environment Type and Requirements 27
6.2 Configurable Settings and Restrictions 27

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Physical Security ... 28
8 Non-Invasive Security ... 29
9 Sensitive Security Parameter Management ... 30
9.1 Storage Areas ... 30
9.2 SSP Input-Output Methods ... 30
9.3 Zeroization Methods ... 30
9.4 SSPs ... 30
9.5 Transitions ... 34
10 Self-Tests ... 35
10.1 Pre-Operational Self-Test ... 35
10.2 Conditional Self-Tests ... 35
10.2.1 Conditional Cryptographic Algorithm Tests ... 37
10.2.2 Conditional Pair-Wise Consistency Tests ... 37
10.3 Periodic Self-Tests ... 37
10.4 Error States ... 37
10.5 Operator Initiation ... 38
11 Life-Cycle Assurance ... 39
11.1 Installation, Initialization and Startup Procedures ... 39
11.2 Administrator Guidance ... 40
12 Mitigation of Other Attacks ... 41
13 Glossary and Abbreviations ... 42
14 References ... 43

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2 List of Tables

Table 1: Security Levels 7
Table 2: Tested Module Identification 9
Table 3: Tested Operational Environments 10
Table 4: Modes of Operation of the Module 10
Table 5: Approved Algorithms 15
Table 6: Vendor Affirmed Algorithms 15
Table 7: Non-Approved Allowed Algorithms with No Security Claimed 16
Table 8: Non-Approved Algorithms, Not Allowed in the Approved Mode of Operation 16
Table 9: Security Function Implementation 17
Table 12: Key Generation 20
Table 13: Key Establishment 20
Table 14: Ports and Interfaces 21
Table 15: Roles 22
Table 16: Approved Services 24
Table 17: Non-Approved Services 25
Table 18: Storage Areas 30
Table 19: SSP Input-Output 30
Table 20: Zeroization Methods 30
Table 21: SSP Information First 32
Table 22: SSP Information Second 34
Table 23: Pre-Operational Self-Tests 35
Table 24: Conditional Self-Tests 37
Table 25: Error States 38

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

3 List of Figures

Figure 1: Block diagram 9

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Copyrights and Trademarks

Amazon is a registered trademark of Amazon Web Services, Inc. or its affiliates.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

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

Table 1 describes the individual security areas of FIPS 140-3, as well as the security levels of those individual areas.

ISO/IEC 24759 Section 6. Subsections	FIPS 140-3 Section Title	Security Level
1	General	1
2	Cryptographic Module Specification	1
3	Cryptographic Module Interfaces	1
4	Roles, Services, and Authentication	1
5	Software/Firmware Security	1
6	Operational Environment	1
7	Physical Security	N/A
8	Non-invasive Security	N/A
9	Sensitive Security Parameter Management	1
10	Self-tests	1
11	Life-cycle Assurance	1
12	Mitigation of Other Attacks	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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

The vendor has provided the non-proprietary Security Policy of the cryptographic module, which was further consolidated into this document by atsec information security together with other vendor-supplied documentation. 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

Module Characteristics: N/A

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

The module components consist of the bcm.o (AWS-LC FIPS 2.0.0), which is dynamically linked to the userspace application during the compilation process.

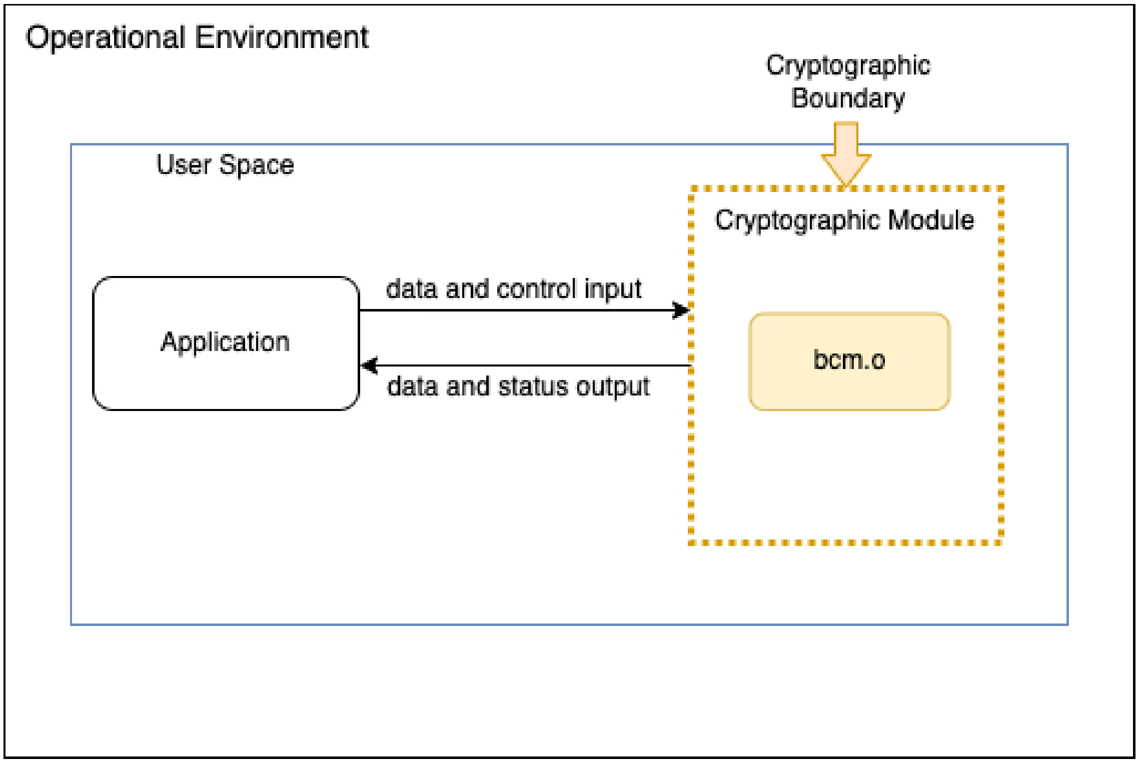

Figure 1: Block diagram

2.2 Operating Environments

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

Package/File Names	Software/ Firmware Version	Integrity Test Implemented
bcm.o	AWS-LC FIPS 2.0.0	HMAC-SHA2-256

Table 2: Tested Module Identification

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processor(s)	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 ®Xeon ® Platinum 8275CL	AES-NI and SHA extensions (PAA)	N/A	AWS-LC FIPS 2.0.0
Amazon Linux 2023					
Ubuntu 22.04					
Amazon Linux 2	Amazon EC2 c5.metal with 192 GiB system memory and Elastic Block Store (EBS) 200 GiB	Intel ®Xeon ® Platinum 8275CL	None		
Amazon Linux 2023					
Ubuntu 22.04					
Amazon Linux 2	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	Neon and Crypto Extension (CE) (PAA)		AWS-LC FIPS 2.0.0
Amazon Linux 2023					
Ubuntu 22.04					
Amazon Linux 2	Amazon EC2 c7g.metal with 128 GiB system memory and Elastic Block Store (EBS) 200 GiB	Graviton3	None		
Amazon Linux 2023					
Ubuntu 22.04					

Table 3: Tested Operational Environments

2.3 Excluded Components

The module does not claim any excluded components.

2.4 Modes of Operation of the Module

Name	Description	Type	Status Indicator
Approved Mode	Automatically entered whenever an approved service is requested.	Approved	Equivalent to the indicator of the requested service.
Non-approved Mode	Automatically entered whenever a non-approved service is requested.	Non-Approved	Equivalent to the indicator of the requested service.

Table 4: Modes of Operation of the Module

Mode change instructions and status indicators:

When the module starts up successfully, after passing a set of cryptographic algorithms self-tests (CASTs) and the pre-operational self-test, 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 Table 15. 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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.5 Algorithms

Approved Algorithms:

Algorithm Name	CAVP Cert Numbers	Algorithm Capabilities	OE (Implementation)	Reference
AES-CBC	A4489, A4493, A4501, A4484, A4487, A4497	Encryption, Decryption using 128,192,256 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM	FIPS 197, SP800-38A
AES-CCM	A4489, A4493, A4501, A4484, A4487, A4497	Authenticated Encryption, Authenticated Decryption, Key Wrapping, Key Unwrapping using 128 bit key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM	FIPS 197, SP800-38C, IG D.G
AES-CMAC	A4489, A4493, A4501, A4487, A4497	Message Authentication Generation 128- or 256-bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM	FIPS 197, SP800-38B

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm Name	CAVP Cert Numbers	Algorithm Capabilities	OE (Implementation)	Reference
AES-CTR	A4489, A4493, A4501, A4484, A4487, A4497	Encryption, Decryption 128,192,256 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM	FIPS 197, SP 800-38A
AES-ECB	A4489, A4490, A4493, A4494, A4496, A4501, A4502, A4503, A4504, A4484, A4485, A4486, A4487, A4488, A4495, A4497, A4498, A4499, A4500	Encryption, Decryption using 128, 192, 256 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, AES_C_GCM, CE, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES, VPAES_GCM **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, AES_C_GCM, CE, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES, VPAES_GCM **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, AES_C_GCM, CE, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES VPAES_GCM **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESNI_AVX, AESNI_ASM, AESASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM, AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESNI_AVX, AESNI_ASM, AESASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM, AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESNI_AVX, AESNI_ASM, AESASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM, AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM	FIPS 197, SP 800-38A
AES-GCM	A4490, A4494, A4496, A4502, A4503, A4504, A4485, A4486, A4488, A4495, A4498, A4499, A4500	Authenticated Encryption (with Internal IV Mode 8.2.2) and Key Wrapping using 128 or 256 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, AES_C_GCM, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES_GCM **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, AES_C_GCM, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES_GCM **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, AES_C_GCM, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES_GCM **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** ASENI_AVX, AESNI_ASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM,	FIPS 197, SP800-38D, IG D.G
AES-GCM		Authenticated Decryption (with external IV) and Key Unwrapping using 128- or 256-bits key		FIPS 197, SP800-38D, IG D.G

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm Name	CAVP Cert Numbers	Algorithm Capabilities	OE (Implementation)	Reference
AES-GMAC		Message Authentication Generation using 128- or 256-bits key	AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** ASENI_AVX, AESNI_ASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM, AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** ASENI_AVX, AESNI_ASM, AESASM_AVX, AES_CLMULNI, AESASM_ASM, AESASM_CLMULNI, AESNI_CLMULNI, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM	FIPS 197, SP800-38D
AES-KW	A4489, A4493, A4501, A4484, A4487, A4497	Key Wrapping, Key Unwrapping using 128, 192, 256 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, BAES_CTASM, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM	FIPS 197, SP800-38F, IG D.G
AES-KWP				
AES-XTS		Encryption, Decryption using 256 bits key		FIPS 197, SP 800-38E
CTR_DRBG	A4489, A4493, A4501, A4484, A4487, A4497	Random Number Generation using AES 256 bits without derivation function or prediction resistance.	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** AES_C, CE, VPAES **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** AES_C, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** AES_C, CE, VPAES **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** AESNI, AESASM, BAES_CTASM	SP800-90Arev1
ECDSA	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Generation using P-224, P-256, P-384, P-521	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	FIPS 186-5 A.2.2 FIPS 186-5 Rejection Sampling; SP800-133rev2 sections 4, 5.1, 5.2
		Key Verification using P-224, P-256, P-384, P-521		FIPS 186-5 for all except FIPS 186-4 for signature verification with SHA-1
ECDSA with SHA2-224, SHA2-256, SHA2-384, SHA2-512		Signature Generation using P-224, P-256, P-384, P-521		

Algorithm Name	CAVP Cert Numbers	Algorithm Capabilities	OE (Implementation)	Reference
ECDSA with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512		Signature Verification using P-224, P-256, P-384, P-521		
HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Authentication Generation using 112-524288 bits key	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	FIPS 198-1
KAS-ECC-SSC ECC Ephemeral Unified scheme	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Shared Secret Computation using P-224, P-256, P-384, P-521		SP800-56ARev3, IG D.F scenario 2(1)
KDA HKDF with HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA-256, HMAC-SHA2-384, HMAC-SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Derivation Derived Key Length: 2048 Shared Secret Length: 224-2048 Increment 8		SP800-56Crev1; SP800-133rev2 section 6.2
KDF TLS (CVL) TLS 1.0/1.1, TLS 1.2 (RFC 7627) with SHA2-256, SHA2-384, SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Derivation		SP800-135rev1; SP800-133rev2 section 6.2
PBKDF with HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Password based key derivation: Iteration Count: 1000-10000 Increment 1 Password Length: 14-128 Increment 1 Salt Length: 128-4096 Increment 8 Key Data Length: 128-4096 Increment 8	**Amazon Linux 2023 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Ubuntu on EC2 bare metal on Amazon Graviton3 AWS Graviton:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Amazon Graviton3:** SHA_ASM, SHA_CE, NEON **Amazon Linux 2 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Amazon Linux 2023 on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3 **Ubuntu on EC2 bare metal on Intel Cascade Lake Xeon Platinum 8275CL:** SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	SP800-132 Option 1a; SP800-133rev2 section 6.2
RSA	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Generation using 2048,3072, 4096 bits key		FIPS 186-5 A.1.3 Random Probable Primes; SP800-133rev2 sections 4, 5.1

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm Name	CAVP Cert Numbers	Algorithm Capabilities	OE (Implementation)	Reference
RSA PKCS#1v1.5 with SHA2-224, SHA2-256, SHA2-384, SHA2-512 RSA PSS with SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/256		Signature Generation using 2048,3072, 4096 bits key		
RSA PKCS#1v1.5 with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512; RSA PSS with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/256		Signature Verification using 1024, 2048, 3072, 4096 bits key.		FIPS 186-5 except FIPS 186-4 for use of SHA-1 and 1024 bit key
SSH KDF (CVL) with AES-128, AES-192, AES-256; SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Key Derivation		SP800-135rev1; SP800-133rev2 section 6.2
SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/256	A4483, A4491, A4492, A4505, A4506, A4507, A4508	Message Digest		FIPS 180-4

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Algorithm Name	Algorithm Capabilities	OE (Implementation)	References
CKG (ECDSA KeyGen)	ECDSA KeyGen (FIPS 186-5): P-224, P-256, P 384, P-521 elliptic curves with 112-256 bits of key strength	Software; OE same as in Table 3	FIPS 186-5, A.2.2 Rejection Sampling; SP 800-133Rev2 section 4 and IG D.H comment 2 (without any V, as described in Additional Comments 2 of IG D.H)
CKG (RSA KeyGen)	RSA KeyGen (FIPS 186-5): 2048, 3072, 4096 bits with 112, 128, 149 bits of key strength.		FIPS 186-5, A.1.3 Random Probable Primes; SP 800-133Rev2 section 4 and IG D.H comment 2 (without any V, as described in Additional Comments 2 of IG D.H)

Table 6: Vendor Affirmed Algorithms

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Non-Approved, Allowed Algorithms:

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:

Algorithm	Caveat	Use/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:

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

Table 8: Non-Approved Algorithms, Not Allowed in the Approved Mode of Operation

2.6 Security Function Implementation

Name	Type	Description	SF Capabilities	Algorithms
KAS-ECC-SSC	KAS	SP800-56Ar3. KAS-ECC-SSC per IG D.F 2 path (1)	Ephemeral Unified scheme Curves: P-224, P-256, P-384, P-521 elliptic curves with 112-256 bits of strength	KAS-ECC-SSC: A4483, A4491, A4492, A4505, A4506, A4507, A4508
AES KW, AES-KWP	KTS	SP 800-38F. KTS (Key Wrapping, Key Unwrapping) per IG D.G	128, 192, 256 bits with 128-256 bits of key strength	AES: A4489, A4493, A4501, A4484, A4487, A4497

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

AES GCM [SP 800-38D]	KTS	SP800-38D. KTS (Key Wrapping, Key Unwrapping) per IG D.G	128, 256 bits with 128 and 256 bits of key strength	AES: A4490, A4494, A4496, A4502, A4503, A4504, A4485, A4486, A4488, A4495, A4498, A4499, A4500
AES CCM [SP 800-38C]	KTS	SP 800-38C. KTS (Key Wrapping, Key Unwrapping) per IG D.G	128 bits with 128 bits of key strength	AES: A4489, A4493, A4501, A4484, A4487, A4497

Table 9: Security Function Implementation

2.7 Algorithm Specific Information

2.7.1 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, and using the referenced APIs explained next.

Scenario 1, TLS 1.2

For TLS 1.2, the module offers the GCM implementation via the functions EVP_aead_aes_128_gcm_tls12() and EVP_aead_aes_256_gcm_tls12(), 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 AES-GCM IV is generated randomly internal to the module using module's approved DRGB. The DRBG receives a LOAD command with entropy obtained from inside the physical perimeter of the operational environment but outside of module's cryptographic boundary. The GCM IV is 96 bits in length and is expected to have 96 bits of entropy.

Scenario 5, TLS 1.3

For TLS 1.3, the module offers the AES-GCM implementation via the functions EVP_aead_aes_128_gcm_tls13() and EVP_aead_aes_256_gcm_tls13(), and uses the context of Scenario 5 of IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of 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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

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.

2.7.2 AES XTS

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

2.7.3 Key Derivation using SP 800-132 PBKDF2

The module provides password-based key derivation (PBKDF2), compliant with SP 800-132. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance 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.

2.7.4 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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

2.7.5 Approved Modulus Sizes for RSA Digital Signature

Following IG C.F, RSA SigGen (FIPS 186-5) and RSA SigVer (FIPS 186-4 and FIPS 186-5) have been CAVP tested with all supported approved RSA modulus lengths (i.e., 1024 (SigVer only), 2048, 3072, 4096). This is documented in the Approved Algorithms table. There are no modulus sizes available in approved services which have not been CAVP tested. The minimum number of the Miller-Rabin tests used in primality testing is consistent with Table B.1 in FIPS 186-5.

2.7.6 Legacy Algorithms

The cryptographic module implements the following cryptographic algorithms for legacy use:

- RSA SigVer (FIPS 186-4) with 1024-bit keys.
- RSA SigVer (FIPS 186-4) with SHA-1.
- ECDSA SigVer (FIPS 186-4) with SHA-1.

2.8 RNG and Entropy

The module provides an SP800-90Arev1-compliant Deterministic Random Bit Generator (DRBG) using CTR_DRBG mechanism with AES-256, without a derivation function, for generation of key components of asymmetric keys, and random number generation. The DRBG receives a LOAD command with entropy obtained from inside the physical perimeter of the operational environment but outside of module's cryptographic boundary. This corresponds to scenario 2 (b) of IG 9.3.A. 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

Name	Type	Properties
ECDSA	CKG	EC: P-224, P-256, P 384, P-521 elliptic curves with 112-256 bits of key strength Method: FIPS 186-5 A.2.2 Rejection Sampling using a DRBG compliant with SP800-90Arev1, per SP800-133Rev2 section 4 (without any V, as described in Additional Comments 2 of IG D.H) and SP800-133Rev2 section 5.1 and 5.2
RSA	CKG	RSA: 2048, 3072, 4096 bits with 112, 128, 149 bits of key strength. Method: FIPS 186-5 A.1.3 Random Probable Primes using a DRBG compliant with SP800-90Arev1, per SP800-133Rev2 section 4 (without any V, as described in Additional Comments 2 of IG D.H) and SP800-133Rev2 section 5.1
KDA HKDF	Key Derivation	Key type: Symmetric key; Security strength: 112-256 bits Method: SP 800-56Cr1; (HMAC) SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 per SP800-133Rev2 section 6.2
PBKDF	Key Derivation	Key type: Symmetric key; Security strength: 112-256 bits Method: option 1a of SP 800-132; (HMAC) SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 per SP800-133Rev2 section 6.2
SSH KDF (CVL)	Key Derivation	Key type: Symmetric key; Security strength: 112-256 bits Method: SP 800-135r1; AES-128, AES-192, AES-256 with SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 per SP800-133Rev2 section 6.2
KDF TLS (CVL) TLS 1.0/1.1, TLS 1.2	Key Derivation	Key type: Symmetric key; Security strength: 112-256 bits Method: SP 800-135r1; MD5 (TLS 1.0/1.1 only), SHA2-256, SHA2-384, SHA2-512 per SP800-133Rev2 section 6.2

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

(RFC 7627)		

Table 10: Key Generation

2.10 Key Establishment

Name	Type	Properties
KAS-ECC-SSC [SP800-56Arev3]	KAS (Shared Secret Computation)	Curves: P-224, P-256, P-384, P-521 elliptic curves with 112-256 bits of key strength Compliant with IG D.F scenario 2(1)
AES GCM [SP 800-38D]	KTS (Key wrapping, Key unwrapping)	128 and 256 bits with 128 and 256 bits of key strength Compliant with IG D.G
AES CCM [SP 800-38C]		128 bits with 128 bits of key strength Compliant with IG D.G
AES KW, AES KWP [SP 800-38F]		128, 192, 256 bits with 128-256 bits of key strength Compliant with IG D.G

Table 11: Key Establishment

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 following key derivation functions for use in the TLS protocol:

- KDF TLS (CVL) TLS 1.0/1.1, TLS 1.2 (RFC 7627)
- KDA HKDF for TLS 1.3

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

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

3 Cryptographic Module Interfaces

As a Software module, the module interfaces are defined as Software or Firmware Module Interfaces (SMFI), and there are no physical ports.

Logical Interface	Data that passes over port/interface
Data Input	API input parameters for data.
Data Output	API output parameters for data.
Control Input	API function calls.
Status Output	API return codes, error message.

Table 12: Ports and Interfaces [1]

---

[1] The control output interface is omitted on purpose because the module does not implement it. The physical ports are not applicable because the module is software only.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not support authentication.

4.2 Roles

The module does not support concurrent operators.

Name	Type	Operator Type	Authentication
Crypto Officer	Role	CO	N/A (Implicitly assumed)

Table 13: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	Roles	SSP Access
Encryption	Encryption	Return value 1 from the function: `FIPS_service_indicator_check_approved()`	AES key, plaintext	Ciphertext	AES-CBC, AES-CTR, AES-ECB, AES-XTS listed in Table 5	CO	AES Key: W, E
Decryption	Decryption		AES key, ciphertext	Plaintext			
Authenticated Encryption	Authenticated Encryption		AES key, IV, plaintext	Ciphertext, MAC tag	AES-CCM, AES-GCM listed in Table 5		AES Key: W, E
Authenticated Decryption	Authenticated Decryption		AES key, ciphertext, MAC tag, IV	Plaintext			
Key wrapping	Encrypting a key		AES key wrapping key, Key to be wrapped	Wrapped key	AES-KW, AES-KWP, AES-CCM, AES-GCM		AES key: W, E
Key unwrapping	Decrypting a key		AES key unwrapping key	Unwrapped key	AES-KW, AES-KWP, AES-CCM, AES-GCM		AES key: W, E
Message Authentication Generation	MAC computation		AES key, message	MAC tag	AES-CMAC, AES-GMAC		AES Key: W, E
			HMAC key, message		HMAC		HMAC Key: W, E
Message Digest	Generating message digest		Message	Message digest	SHA		N/A
Random Number Generation	Generating random numbers		Output length	Random bytes	CTR_DRBG		Entropy Input: W, E
							DRBG Seed: G, E
							DRBG Internal State (V, Key): G, E
Key Generation	Generating key pair		Modulus size	Module generated RSA public key, Module generated RSA private key	RSA listed in Table 5, CKG		Module generated RSA Public Key: G, R
							Module generated RSA Private Key: G, R
							Intermediate Key Generation Value: G, E, Z

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	Roles	SSP Access
			Curve	Module generated EC public key, Module generated EC private key	ECDSA listed in Table 5, CKG		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		EC Public key	Success/ error	ECDSA listed in Table 5		EC Public Key: W, E
Signature Generation	Generating signature		Message, EC private key or RSA private key	Digital signature	RSA, ECDSA listed in Table 5		EC Private Key: W, E RSA Private Key: W, E
Signature Verification	Verifying signature		Signature, EC public key or RSA public key	Digital signature verification result	RSA, ECDSA listed in Table 5		EC Public Key: W, E RSA Public Key: W, E
Shared Secret Computation	Calculating the Shared Secret		EC public key, EC private key	Shared Secret	KAS-ECC-SSC		EC Public Key: W, E EC Private Key: W, E Shared Secret: G, R
Key Derivation	Deriving Keys		TLS Pre-Master Secret	TLS Master secret	TLS KDF (CVL) TLS 1.0/1.1, TLS 1.2		TLS Pre-Master Secret: W, E TLS Master Secret: G
			TLS Master Secret	TLS Derived Key (AES/HMAC)	TLS KDF (CVL) TLS 1.0/1.1, TLS 1.2 (RFC 7627)		TLS Master Secret: E TLS Derived Key (AES/HMAC): G, R
			Password, salt, iteration count	PBKDF Derived Key	PBKDF2		PBKDF Derived Key: G, R Password: W, E
			Shared Secret, Key Length, Digest	KDA HKDF Derived Key	KDA HKDF		KDA HKDF Derived Key: G, R Shared Secret: W, E
			Shared Secret, Key Length	SSH KDF Derived Key	SSH KDF		SSH KDF Derived Key: G, R Shared Secret: W, E
Zeroization	Zeroize PSP in volatile memory	N/A	SSP	N/A	None		All SSPs: Z
On-Demand Self-test	Initiate power-on self-tests by reset		N/A	Pass or fail	AES, HMAC, SHA, CTR_DRBG, RSA, ECDSA, KAS-ECC-SSC, TLS KDF (CVL) TLS 1.0/1.1, TLS 1.2, KDA HKDF, PBKDF2		N/A
On-Demand Integrity Test	Initiate integrity test on-demand		N/A		HMAC-SHA2-256		N/A

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Indicator	Inputs	Outputs	Security Functions	Roles	SSP Access
Show Status	Show status of the module state		N/A	Module status	N/A		N/A
Show Version	Show the version of the module using `awslc_version_string`		N/A	Module name and version	N/A		N/A

Table 14: Approved Services

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

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

Service	Description	Algorithms Accessed	Role	Indicator
Encryption	Encryption	AES listed in Table 8	CO	Return value 0 from the function FIPS_ service_ indicator_ check_ approved()
Decryption	Decryption			
Message Authentication Generation	MAC computation	AES GMAC and HMAC listed in Table 8		
Message Digest	Generating message digest	MD4, MD5 outside TLS 1.0 usage, SHAKE, SHA-3, RIPEMD-160		
Signature Generation	Generating signature	Using SHA-1, SHAKE, SHA-3		

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Service	Description	Algorithms Accessed	Role	Indicator
		RSA listed in Table 8, Curve secp256k1		
Signature Verification	Verifying signature	RSA listed in Table 8, Curve secp256k1		
Key Generation	Generating key pair	RSA or ECDSA listed in Table 8		
Shared Secret Computation	Calculating shared secret	Diffie-Hellman, Curve secp256k1		
Key Derivation	Deriving TLS keys	TLS KDF listed in Table 8		
Key Encapsulation	Decrypting a key	RSA		
Key Un-encapsulation	Encrypting a key	RSA		
Encryption Primitive	Asymmetric encryption	RSA with PKCS#1 v1.5 and OAEP padding		

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

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

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

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 Integrity Test

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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: The module operates in a modifiable operational environment. The module runs on a commercially available general-purpose operating system executing on the hardware specified in section 2.

How requirements are satisfied: The module should 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 Configurable 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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

7 Physical Security

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

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

8 Non-Invasive Security

The module claims no non-invasive security techniques.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

9 Sensitive Security Parameter 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 16: Storage Areas

9.2 SSP Input-Output Methods

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

Table 17: SSP Input-Output

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 Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free Cipher Handle	Zeroizes the SSPs contained within the cipher handle.	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	By calling the appropriate zeroization functions: OpenSSL_cleanse, EVP_CIPHER_CTX_cleanup, EVP_AEAD_CTX_zero, HMAC_CTX_cleanup, CTR_DRBG_clear, RSA_free, EC_KEY_free
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 18: Zeroization Methods

9.4 SSPs

Name	Description	Size	Strength	Type	Generation	Established By
AES Key	AES key used for encryption, decryption, and computing MAC tags	128, 192, 256 bits	128-256 bits of strength	Symmetric key	N/A	N/A

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size	Strength	Type	Generation	Established By
HMAC Key	HMAC key for Message Authentication Generation	112-524288 bits	112-256 bits of strength	Authentication key	N/A	N/A
Entropy Input (per IG D.L)	Entropy input used to seed the DRBGs	256 bits	256 bits of strength	Entropy	N/A	N/A
DRBG Seed (per IG D.L)	DRBG seed derived from entropy input as defined in SP 800-90Ar1	256 bits	256 bits of strength	DRBG seed	CTR_DRBG (according to SP800-90Arev1)	N/A
DRBG Internal State (V, Key) (per IG D.L)	Internal state of CTR_DRBG	256 bits	256 bits of strength	Internal state	CTR_DRBG (derived from DRBG seed according to SP800-90Ar1)	N/A
RSA Public Key	RSA public key used for signature verification	1024, 2048, 3072, 4096 bits	80-150 bits of strength	Public key	N/A	N/A
RSA Private Key	RSA private key used for signature generation	2048, 3072, 4096 bits	112-150 bits of strength	Private key		N/A
Module generated RSA Public Key	RSA public key generated by the module		112-50 bits of strength	Public key	RSA (generated according to FIPS 186-5) DRBG (for generation of random values)	
Module generated RSA Private Key	RSA private key generated by the module		112-150 bits of strength	Private key		
EC Public Key	EC public key used for key verification, signature verification, shared secret computation	P-224, P-256, P-384, P-521	112-256 bits of strength	Public key	N/A	N/A
EC Private Key	EC private key used for signature generation, shared secret computation			Private key		N/A
Module generated EC Public Key	EC public key generated by the module			Public key	ECDSA (generated according to FIPS 186-5) DRBG (for generation of random values)	N/A
Module generated EC Private Key	EC private key generated by the module			Private key		N/A
Shared Secret	Shared Secret generated by KAS-ECC-SSC			Shard secret	N/A	KAS-ECC-SSC (established according to SP800-56Arev3)
TLS Pre-Master Secret	TLS Pre-Master secret used for deriving the TLS Master Secret	P-224, P-256, P-384, P-521	112-256 bits	TLS pre-master secret	N/A	N/A
TLS Master Secret	TLS Master secret used for deriving the TLS Derived Key	384 bits	112-256 bits	TLS master secret	KDF TLS (CVL) TLS 1.0/1.1, TLS 1.2 (RFC 7627) (derived	N/A

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Description	Size	Strength	Type	Generation	Established By
TLS Derived key (AES/HMAC)	TLS Derived Key from TLS Master Secret	AES: 128-256 bits HMAC: 112 to 256 bits	AES: 128-256 bits of strength HMAC: 112-256 bits of strength	Symmetric key	according to SP800-135rev1)	N/A
KDA HKDF derived key	KDA HKDF derived key	112 to 2048 bits	112-256 bits of strength	Symmetric key	KDA HKDF (derived according to SP800-56Crev1)	N/A
SSH KDF derived key	SSH KDF derived key	112 to 256 bits			SSH KDF (CVL) (derived according to SP800-135rev1)	
PBKDF derived key	PBKDF derived key	112–4096 bits			PBKDF (derived according to SP800-132)	
Password	Password for PBKDF	112-1024 bits	N/A	Password	N/A	N/A
Intermediate Key Generation Value	Intermediate key generation value	224-4096 bits	112-256 bits of strength	Intermediate value	CKG	N/A

Table 19: SSP Information First

Name	Used By	Inputs/Outputs	Storage	Zeroization	Category	Related SSPs
AES Key	Encryption, Decryption, Authenticated Encryption, Authentication Decryption, Key wrapping, Key unwrapping, Message Authentication Generation	API input parameters (input)	RAM	Free Cipher Handle, Module Reset	CSP	None
HMAC Key	Message Authentication Generation	API input parameters (input)			CSP	None
Entropy Input (per IG D.L)	Random Number Generation	API input parameters (input)		Automatically	CSP	DRBG Seed
DRBG Seed (per IG D.L)	Random Number Generation	N/A			CSP	Entropy Input, DRBG Internal State (V, Key)
DRBG Internal State (V, Key) (per IG D.L)	Random Number Generation	N/A		Free Cipher Handle, Module Reset	CSP	DRBG Seed
RSA Public Key	Signature Verification	API input parameters (input)			PSP	RSA Private Key
RSA Private Key	Signature Generation				CSP	RSA Public Key

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Used By	Inputs/Outputs	Storage	Zeroization	Category	Related SSPs
Module generated RSA Public Key	N/A	API output parameters (output)			PSP	Module generated RSA Private Key, Intermediate Key Generation Value
Module generated RSA Private Key	N/A				CSP	Module generated RSA Public Key, Intermediate Key Generation Value
EC Public Key	Key Verification, Signature Verification, Shared Secret Computation	API input parameters (input)			PSP	EC Private Key, Shared Secret
EC Private Key	Signature Generation, Shared Secret Computation				CSP	EC Public Key, Shared Secret
Module generated EC Public Key	EC Public Key generated by the module	API output parameters (output)			PSP	Module generated EC Private Key, Intermediate Key Generation Value
Module generated EC Private Key	EC Private Key generated by the module				CSP	Module generated EC Public Key, Intermediate Key Generation Value
Shared Secret	Key Derivation	API output parameters (output)			CSP	EC Public Key, EC Private Key
TLS Pre-Master Secret	Key Derivation	API input parameters (input)			CSP	TLS Master Secret
TLS Master Secret	Key Derivation	N/A			CSP	TLS Pre-Master Secret, TLS Derived Key (AES/HMAC)
TLS Derived Key (AES/HMAC)	N/A	API output parameters (output)			CSP	TLS Master Secret
KDA HKDF Derived Key					CSP	Shared Secret
SSH KDF Derived Key					CSP	Shared Secret
PBKDF Derived Key					CSP	Password

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Name	Used By	Inputs/Outputs	Storage	Zeroization	Category	Related SSPs
Password		API input parameters (input)			CSP	PBKDF Derived Key
Intermediate Key Generation Value	Key Generation	N/A		Automatically	CSP	Module generated RSA Private Key, Module generated RSA Public Key, Module generated EC Private Key, Module generated EC Public Key

Table 20: SSP Information Second

9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting January 1, 2030.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

10 Self-Tests

10.1 Pre-Operational Self-Test

Algorithm	Implementation	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256	SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	SHA2-256	Message Authentication	Software Integrity	Module becomes operational	N/A

Table 21: 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 includes the self-test for HMAC-SHA2-256 algorithm used in the pre-operational self-test.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Type	Indicator	Details	Condition	Coverage	Coverage Notes
AES CBC **AES GCM** AES_C, AES_C_GCM, AESNI, AESNI_AVX, AESNI_ASM, AESAESM, AESASM_AVX, AESASM_CLMULNI, AESASM_ASM, CE, CE_GCM_UNROLL8_EOR3, CE_GCM, VPAES, VPAES_GCM, AESNI_CLMULNI, BAES_CTASM, BAES_CTASM_AVX, BAES_CTASM_CLMULNI, BAES_CTASM_ASM	128-bit AES key	Encrypt KAT for CBC	CAST	Module is operational	Encrypt	Power up	Self	N/A
		Decrypt KAT for CBC			Decrypt		Self and ECB, KW, KWP, XTS (all implementations)	IG 10.3.A, resolution 1.c
		Encrypt KAT for GCM			Encrypt		Self and CCM, CMAC, CTR, ECB, GMAC, KW, KWP, XTS all implementations)	IG 10.3.A, resolution 1.d.(i)
		Decrypt KAT for GCM			Decrypt		Self	N/A
SHA-1 **SHA2-256** **SHA2-512** SHA_CE, SHA_ASM, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	N/A	SHA-1 KAT	CAST		Message digest	Power up	Self	N/A
		SHA2-256 KAT					Self and SHA2-224 all implementations)	IG 10.3.A, resolution 2
							SSH KDF (all implementations)	IG 10.3.A, resolution 12, note 18
		SHA2-512					Self and SHA2-	IG 10.3.A,

Algorithm or Test	Test Properties	Test Method	Type	Indicator	Details	Condition	Coverage	Coverage Notes
		KAT					384, SHA2-512/256 (all implementations)	resolution 2
HMAC SHA_CE, SHA_ASM, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	SHA2-256	HMAC KAT	CAST		Message authentication	Power up	Self and HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-SHA2-512/256	IG 10.3.A resolution 5
CTR_DRBG AES_C, AESNI, AESASM, AESASM_AVX, CE, VPAES, BAES_CTASM	AES 256	CTR_DRBG KAT	CAST		Seed Generation	Power up	Self	N/A
	N/A	SP800-90Ar1 Section 11.3 Health Test			Seed Generation	Power up	Self	N/A
ECDSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	P-256 Curve and SHA2-256	Sign KAT	CAST		Sign	Signature Generation or Key Generation service request	Self	N/A
ECDSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	P-256 Curve and SHA2-256	Verify KAT			Verify	Signature verification or Key Generation service request	Self	N/A
KAS-ECC-SSC C	P-256 Curve	Z computation			Shared secret computation	Shared secret computation request	Self	N/A
ECDSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	Respective Curve and SHA2-256	Signature generation and verification	PCT		Sign and Verify	Key generation	Self and KAS-ECC-SSC PCT	IG 10.3.A additional comment 1.
KDF TLS (CVL) SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	SHA2-256	TLS 1.2 KAT	CAST		Key derivation	Power up	Self	N/A
KDA HKDF SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	HMAC-SHA2-256	KAT	CAST		Key derivation	Power up	Self	N/A
PBKDF2 SHA_ASM, SHA_CE,	HMAC-SHA2-256	KAT	CAST		Key derivation	Power up	Self	N/A

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

Algorithm or Test	Test Properties	Test Method	Type	Indicator	Details	Condition	Coverage	Coverage Notes
NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3								
RSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Sign KAT	CAST		Sign	Signature Generation or Key Generation service request	Self	N/A
RSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	PKCS#1 v1.5 with 2048 bit key and SHA2-256	Verify KAT	CAST		Verify	Signature Verification or Key Generation service request	Self	N/A
RSA SHA_ASM, SHA_CE, NEON, SHA_SHANI, SHA_AVX2, SHA_AVX, SHA_SSSE3	SHA2-256 and respective keys	Signature generation and verification	PCT		Sign and Verify	Key generation	Self	N/A

Table 22: Conditional Self-Tests

10.2.1 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.

10.2.2 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-Tests

The module does not support periodic self-tests.

10.4 Error States

Name	Description	Condition	Recovery Method	Status Indicator
Error	The library is aborted with SIGABRT signal. Module is no longer operational the data output	Pre-operational test failure	Module reset	Error message is output on the stderr and then the module is aborted.
		Conditional test failure	Module reset	For CAST failure, an error message is output on the stderr and then the module is aborted. For PCT failure, an error message is output in the error queue and then

Name	Description	Condition	Recovery Method	Status Indicator
	interface is inhibited			the module generates new key, If the PCT still does not pass, eventually the module will be aborted after 5 tries.

Table 23: Error States

If the module fails any of the self-tests, the module enters the error state. To recover from the Error state, the module needs to be rebooted.

10.5 Operator Initiation

The software integrity tests and the CASTs for AES, SHA, DRBG, 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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

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 Table 3).

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 -DCMAKE_BUILD_TYPE=Release -DBUILD_SHARED_LIBS=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 -DCMAKE_BUILD_TYPE=Release -DBUILD_SHARED_LIBS=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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

12 Mitigation of Other Attacks

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.

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

13 Glossary and Abbreviations

AES	Advanced Encryption Standard
AESNI	Advanced Encryption Standard New Instructions
CAVP	Cryptographic Algorithm Validation Program
CAST	Cryptographic Algorithm Self-Test
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
KW	AES Key Wrap
KWP	AES Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
OS	Operating System
PAA	Processor Algorithm Acceleration
PCT	Pair-Wise Consistency Test
PR	Prediction Resistance
PSP	Public Security Parameter
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

14 References

FIPS140-3 **FIPS PUB 140-3 - Security Requirements for Cryptographic Modules**
March 2019
https://doi.org/10.6028/NIST.FIPS.140-3

FIPS140-3_IG **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**
August 2023
https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements

FIPS180-4 **Secure Hash Standard (SHS)**
March 2012
http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS186-4 **Digital Signature Standard (DSS)**
July 2013
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS186-5 **Digital Signature Standard (DSS)**
February 2023
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf

FIPS197 **Advanced Encryption Standard**
November 2001
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1 **The Keyed Hash Message Authentication Code (HMAC)**
July 2008
http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

PKCS#1 **Public Key Cryptography Standards (PKCS) #1: RSA Cryptography**
Specifications Version 2.1
February 2003
http://www.ietf.org/rfc/rfc3447.txt

SP800-38A **Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques**
December 2001
http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

SP800-38B **NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**
May 2005
http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

SP800-38C **NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.

SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP800-56Arev3	**NIST Special Publication 800-56A Revision 2 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** May 2013 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf
SP800-90Arev1	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-131Arev1	**NIST Special Publication 800-131A Revision 1- Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths** November 2015 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf
SP800-133rev2	**NIST Special Publication 800-133rev2 - Recommendation for Cryptographic** **Key Generation** June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP800-135rev1	**NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf

© 2024 Amazon Web Services, Inc., atsec information security.
This document can be reproduced and distributed only whole and intact, including this copyright notice.