Amazon Web Services, Inc.

Amazon Linux 2023 Libgcrypt Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.2
Last update: 2025-02-05

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 1.10.2- 752a14d13e4ce9c0 of the Amazon Linux 2023 Libgcrypt Cryptographic Module. It has a one-to- one mapping to the SP 800-140Brev1 starting with Section B.2.1 named "General" that maps to Section 1 General in this document and ending with Section B.2.12 named "Mitigation of other attacks" that maps to Section 12 Mitigation of Other Attacks in this document.

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

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Amazon Linux 2023 Libgcrypt Cryptographic Module (hereafter referred to as "the module") is a Software multi-chip standalone cryptographic module. The module is a software library implementing general purpose cryptographic algorithms. The module provides cryptographic services to applications running in the user space of the underlying operating system through an application program interface (API).

Module Type: Software

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The module is implemented as shared library / binary file; as shown in the diagram below, the shared library file constitutes the cryptographic boundary.

Tested Operational Environment's Physical Perimeter (TOEPP):

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

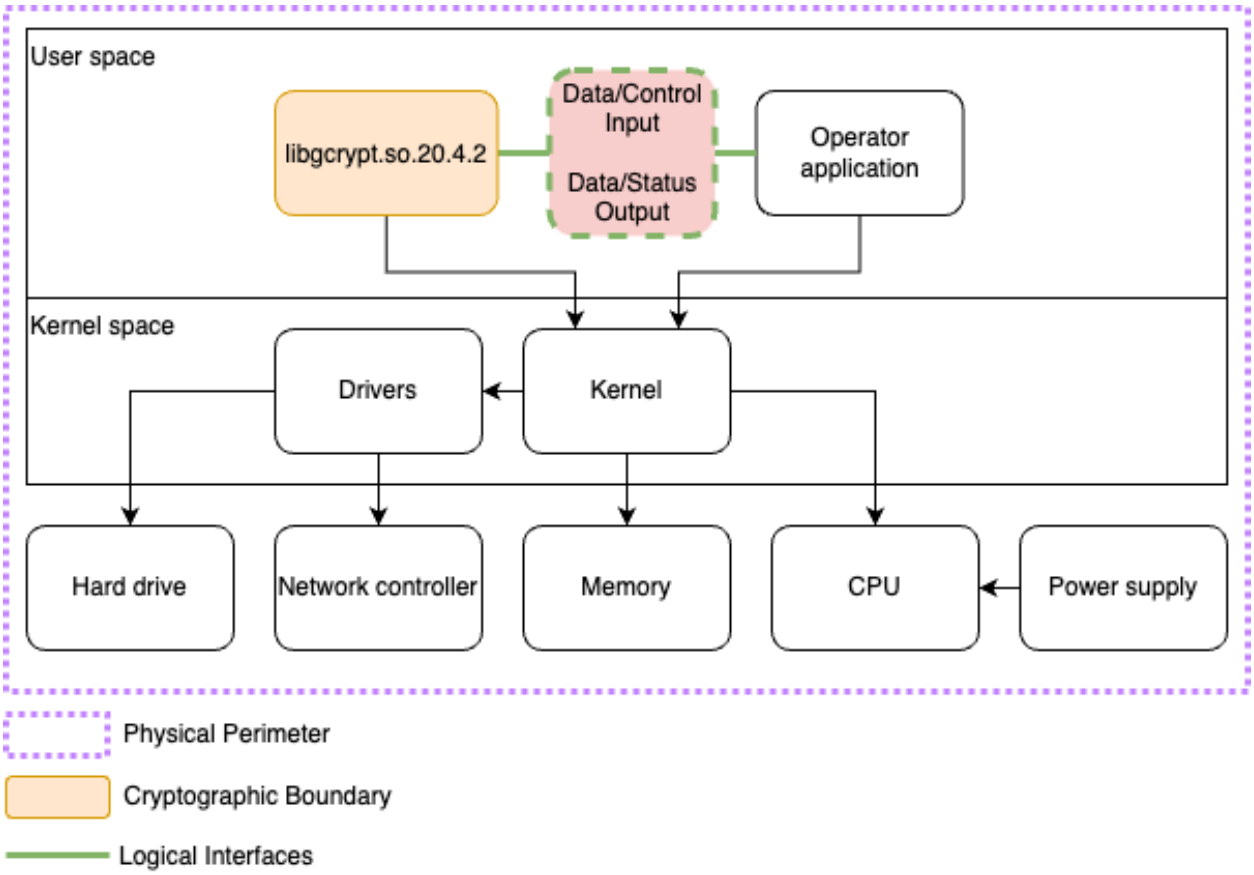

Figure 1: Block Diagram

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
libgcrypt.so.20.4.2 on Amazon Linux 2023 with AWS Graviton3	1.10.2-e4ddf5ed7abe8d45	N/A	HMAC-SHA-256

Package or File Name	Software/ Firmware Version	Features	Integrity Test
libgcrypt.so.20.4.2 on Amazon Linux 2023 with Intel Xeon Platinum 8375C	1.10.2-e4ddf5ed7abe8d45	N/A	HMAC-SHA-256
libgcrypt.so.20.4.2 on SnowOS 1.0 with AMD EPYC 7702	1.10.2-e4ddf5ed7abe8d45	N/A	HMAC-SHA-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 2023	EC2 c7g.metal	AWS Graviton3	Yes	N/A	1.10.2-e4ddf5ed7abe8d45
Amazon Linux 2023	EC2 c6i.metal	Intel Xeon Platinum 8375C	Yes	N/A	1.10.2-e4ddf5ed7abe8d45
SnowOS 1.0	AWS Snowball	AMD EPYC 7702	Yes	N/A	1.10.2-e4ddf5ed7abe8d45

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

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

N/A for this module.

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

2.3 Excluded Components

There are no components excluded from the module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator 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 List and Description

Mode Change Instructions and Status:

When the module starts up successfully, after passing the pre-operational self-test and cryptographic algorithms self-tests, 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 table above. Refer to Section 4 Roles, Services, and Authentication for details on the service indicators provided by the module that identify 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	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A4597	Key Length - 128, 192, 256	SP 800-38C
AES-CCM	A4598	Key Length - 128, 192, 256	SP 800-38C
AES-CCM	A4600	Key Length - 128, 192, 256	SP 800-38C
AES-CCM	A4601	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A4597	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CMAC	A4598	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CMAC	A4600	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B

Algorithm	CAVP Cert	Properties	Reference
AES-CMAC	A4601	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-KW	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A4597	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-OFB	A4598	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-OFB	A4600	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-OFB	A4601	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4597	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
AES-XTS Testing Revision 2.0	A4598	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
AES-XTS Testing Revision 2.0	A4600	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
AES-XTS Testing Revision 2.0	A4601	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A4597	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
Counter DRBG	A4598	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
Counter DRBG	A4600	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
Counter DRBG	A4601	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A4597	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A4598	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A4600	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A4601	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A4602	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4597	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4598	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4600	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4601	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4602	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4597	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4598	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4600	Component - No Curve - P-224, P-256, P-384, P-521	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	
ECDSA SigGen (FIPS186-4)	A4601	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4602	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4597	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4598	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4600	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4601	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-4)	A4602	Component - No Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	FIPS 186-4
Hash DRBG	A4597	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
Hash DRBG	A4598	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
Hash DRBG	A4600	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
Hash DRBG	A4601	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
Hash DRBG	A4602	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A4597	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A4598	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A4600	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A4601	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A4602	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A4596	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA-1	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA-1	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A4599	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA-1	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA-1	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA-1	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4600	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4601	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA3-256	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4597	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4598	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4602	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
PBKDF	A4597	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
PBKDF	A4598	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
PBKDF	A4600	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
PBKDF	A4601	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
PBKDF	A4602	Iteration Count - Iteration Count: 1000-10000000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-4)	A4597	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA KeyGen (FIPS186-4)	A4598	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
		Primality Tests - Table C.2 Private Key Format - Standard	
RSA KeyGen (FIPS186-4)	A4600	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA KeyGen (FIPS186-4)	A4601	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA KeyGen (FIPS186-4)	A4602	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4
RSA SigGen (FIPS186-4)	A4597	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-4)	A4598	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-4)	A4600	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-4)	A4601	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-4)	A4602	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-2)	A4597	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-2)	A4598	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-2)	A4600	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-2)	A4601	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-2)	A4602	Signature Type - PKCS 1.5, PKCSPSS Modulo - 1024, 1536	FIPS 186-4
RSA SigVer (FIPS186-4)	A4597	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A4598	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A4600	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A4601	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A4602	Signature Type - PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
SHA-1	A4596	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4599	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA-1	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-224	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-384	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/224	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4600	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512/256	A4601	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-224	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-224	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A4597	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A4598	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A4602	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202

Algorithm	CAVP Cert	Properties	Reference
SHAKE-128	A4597	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-128	A4598	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-128	A4602	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4597	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4598	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4602	Output Length - Output Length: 16-65536 Increment 8	FIPS 202

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Capabilities:Key generation RSA:2048, 3072, 4096 (112, 128, 149 bits) ECDSA:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	Amazon Linux 2023 Libgcrypt Cryptographic Module (Full Acceleration)	FIPS 186-4, SP 800-133rev2 Section 5.1
CKG	Capabilities:Key generation RSA:2048, 3072, 4096 (112, 128, 192 bits) ECDSA:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	Amazon Linux 2023 Libgcrypt Cryptographic Module (No Acceleration)	FIPS 186-4, SP 800-133rev2 Section 5.1
CKG	Capabilities:Key generation RSA:2048, 3072, 4096 (112, 128, 149 bits) ECDSA:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	Amazon Linux 2023 Libgcrypt Cryptographic Module (AESNI AVX)	FIPS 186-4, SP 800-133rev2 Section 5.1
CKG	Capabilities:Key generation RSA:2048, 3072, 4096 (112, 128, 149 bits) ECDSA:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	Amazon Linux 2023 Libgcrypt Cryptographic Module (SSSE3)	FIPS 186-4, SP 800-133rev2 Section 5.1

Name	Properties	Implementation	Reference
CKG	Capabilities:Key generation RSA:2048, 3072, 4096 (112, 128, 149 bits) ECDSA:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	Amazon Linux 2023 Libgcrypt Cryptographic Module (SHLD)	FIPS 186-4, SP 800-133rev2 Section 5.1

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:

N/A for this module.

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

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
MD5	Message Digest
ECDH	Shared Secret Computation
AES-GCM, AES-GCM-SIV, AES-OCB, AES-EAX	Symmetric encryption; Symmetric decryption
RSA	Signature generation primitives; Signature verification primitives; Encryption primitives; Decryption primitives
RSA with non-approved public key flags	Key generation; Signature generation; Signature verification
ECDSA	Signature generation primitives; Signature verification primitives
ECDSA with non-approved public key flags	Key generation; Signature generation; Signature verification

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Key wrapping using AES CCM	KTS-Wrap	Key wrapping using AES CCM	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM
Key unwrapping using AES CCM	KTS-Wrap	Key unwrapping using AES CCM	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM
Key wrapping using AES KW	KTS-Wrap	Key wrapping using AES KW	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-KW AES-KW AES-KW AES-KW
Key unwrapping using AES KW	KTS-Wrap	Key wrapping using AES KW	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-KW AES-KW AES-KW AES-KW
Encryption with AES	BC-UnAuth	Encryption using AES	XTS mode key sizes and strength:128, 256 bits keys with 128, 256 bits of key strength, respectively Other modes key sizes and strength:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CBC AES-CBC AES-CBC AES-CBC AES-OFB AES-OFB AES-OFB AES-OFB AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CTR

Name	Type	Description	Properties	Algorithms
				AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB AES-ECB AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Authenticated encryption with AES	BC-Auth	Authenticated encryption using AES	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM AES-KW AES-KW AES-KW AES-KW
Decryption with AES	BC-UnAuth	Decryption using AES	XTS mode key sizes and strength:128, 256 bits keys with 128, 256 bits of key strength, respectively Other modes key sizes and strength:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CBC AES-CBC AES-CBC AES-CBC AES-OFB AES-OFB AES-OFB AES-OFB AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CTR

Name	Type	Description	Properties	Algorithms
				AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB AES-ECB AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Authenticated decryption with AES	BC-Auth	Authenticated decryption using AES	Key:128, 192, 256 bits keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM AES-KW AES-KW AES-KW AES-KW
Key Pair Generation with RSA	AsymKeyPair-KeyGen	Key pair generation for RSA	Mode:B.3.3. Random Probable Primes Modulus:2048, 3072, 4096 (112, 128, 149 bits)	RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4)
Key Pair Generation with ECDSA	AsymKeyPair-KeyGen	Key pair generation for ECDSA	Mode:B.4.2 Testing Candidates Curves:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4)

Name	Type	Description	Properties	Algorithms
				ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4)
Public Key Verification with ECDSA	AsymKeyPair-KeyVer	Verify public key for ECDSA	Curves:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4)
Signature Generation with RSA	DigSig-SigGen	Digital signature generation using RSA	Padding:PKCS#1 v1.5, PSS Keys:2048, 3072, 4096 bits (112, 128, 149 bits) Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4)
Signature Verification with RSA	DigSig-SigVer	Digital signature verification using RSA	Padding:PKCS#1 v1.5, PSS Keys:1024, 1536, 2048, 3072, 4096 (80, 96, 112, 128, 149 bits) Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-2) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer

Name	Type	Description	Properties	Algorithms
				(FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4)
Signature Generation with ECDSA	DigSig-SigGen	Digital signature generation using ECDSA	Curves:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits) Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4)
Signature Verification with ECDSA	DigSig-SigVer	Digital signature verification using ECDSA	Curves:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits) Hashes:SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4)
Hashes	SHA	Compute a message digest using Secure Hash Algorithms		SHA-1 SHA-1 SHA-1 SHA-1 SHA-1 SHA-1 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-256 SHA2-256

Name	Type	Description	Properties	Algorithms
				SHA2-256 SHA2-256 SHA2-256 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA3-224 SHA3-224 SHA3-224 SHA3-256 SHA3-256 SHA3-256 SHA3-384 SHA3-384 SHA3-384 SHA3-512 SHA3-512 SHA3-512
Extendable Output Functions	XOF	Compute a message digest from XOFs		SHAKE-128 SHAKE-128 SHAKE-128 SHAKE-256 SHAKE-256 SHAKE-256

Name	Type	Description	Properties	Algorithms
Message Authentication Code	MAC	Compute MAC tags using AES-based CMAC or HMAC	Keys:112-256 bits	HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/256 HMAC-SHA2-512/256

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA3-224 HMAC-SHA3-224 HMAC-SHA3-224 HMAC-SHA3-256 HMAC-SHA3-256 HMAC-SHA3-256 HMAC-SHA3-384 HMAC-SHA3-384 HMAC-SHA3-384 HMAC-SHA3-512 HMAC-SHA3-512 HMAC-SHA3-512 AES-CMAC AES-CMAC AES-CMAC AES-CMAC
Random Number Generation with DRBG	DRBG	Random number generation using DRBG	Compliance:Compliant with SP 800-90Arev1	Counter DRBG Counter DRBG Counter DRBG Counter DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG Hash DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG HMAC DRBG
Key Derivation with PBKDF	PBKDF	Key derivation using PBKDF	Derived keys:112-256 bits	PBKDF PBKDF PBKDF

Name	Type	Description	Properties	Algorithms
				PBKDF PBKDF

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES XTS

The AES algorithm in XTS mode can be only used for the cryptographic protection of data on storage devices, as specified in SP 800-38E. The length of a single data unit encrypted with the XTS-AES shall not exceed 2^{20} AES blocks, that is 16MB of data.

To meet the requirement stated in IG C.I, the module implements a check that ensures, before performing any cryptographic operation, that the two AES keys used in AES XTS mode are not identical.

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

2.7.2 Key Derivation using SP 800-132 PBKDF

The module provides password-based key derivation (PBKDF), 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 Master Key (MK) shall not be used for other purposes. The module accepts length of the MK or DPK of 112 bits or more.
- A portion of the salt, with a length of at least 128 bits, shall be generated randomly using the SP 800-90Arev1 DRBG,
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value accepted by the module is 1000.
- Passwords or passphrases, used as an input for the PBKDF, shall not be used as cryptographic keys.
- The length of the password or passphrase shall be of at least 8 characters, and shall consist of lower-case, upper-case, and numeric characters. Assuming the worst-case scenario of all digits, the probability is estimated to be at most 10^{-8}. Combined with the minimum iteration count as described above, this provides an acceptable trade-off between user experience and security against brute-force attacks.

The calling application shall also observe the rest of the requirements and recommendations specified in SP 800-132.

2.8 RBG and Entropy

Cert Number	Vendor Name
E124	Amazon Web Services, Inc.

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Userspace CPU Time Jitter RNG Entropy Source version 3.4.0	Non-Physical	Amazon Linux 2023 on EC2 c7g.metal; Amazon Linux 2023 on EC2 c6i.metal; SnowOS 1.0 on AWS Snowball	64 bits	Full entropy	SHA3-256 (A4551); HMAC_DRBG (A4551)

Table 10: Entropy Sources

The module provides an SP 800-90Arev1-compliant Deterministic Random Bit Generator (DRBG) for creation of key components of asymmetric keys, and random number generation. This entropy source is located within the module's physical perimeter but outside of the module's cryptographic boundary. The module obtains 384 bits to seed the DRBG, and 256 bits to reseed it.

The seeding (and automatic reseeding) of the DRBG is done with getrandom().

The DRBG supports the Hash_DRBG, HMAC_DRBG and CTR_DRBG mechanisms. The DRBG is initialized during module initialization; the module loads by default the DRBG using the HMAC_DRBG mechanism with SHA-256 and without prediction resistance. A different DRBG mechanism can be chosen by invoking the gcry_control(GCRYCTL_DRBG_REINIT) function.

The module performs the DRBG health tests as defined in Section 11.3 of SP 800-90Arev1.

2.9 Key Generation

The module provides an SP 800-90Arev1-compliant Deterministic Random Bit Generator (DRBG) for the creation of key components of asymmetric keys, and random number generation.

The Cryptographic Key Generation (CKG) methods implemented in the module for Approved services in the approved mode are compliant with Section 5.1 of SP 800-133rev2.

For generating RSA and ECDSA keys the module implements asymmetric key generation services compliant with FIPS 186-4. A seed (i.e., the random value) used in asymmetric key generation is directly obtained from the SP 800-90Arev1 DRBG.

Additionally, the module implements key derivation with PBKDF2 using option 1a, compliant with SP800-132.

2.10 Key Establishment

The module implements the SSP transport methods as specified in the *Security Function Implementations* table.

2.11 Industry Protocols

The module does not support any industry protocols listed within the publication of SP 800-135rev1. Therefore, this section is not applicable.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
As a software-only module, the module does not have physical ports. The operator can only interact with the module through the API provided by the module. Thus, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.	Data Input	API input parameters for data
As a software-only module, the module does not have physical ports. The operator can only interact with the module through the API provided by the module. Thus, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.	Data Output	API output parameters for data
As a software-only module, the module does not have physical ports. The operator can only interact with the module through the API provided by the module. Thus, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.	Control Input	API function calls, API input parameters for control input
As a software-only module, the module does not have physical ports. The operator can only interact with the module through the API provided by the module. Thus, the physical ports are interpreted to be the physical ports of the hardware platform on which the module runs.	Status Output	API return codes, API output parameters for status output

Table 11: Ports and Interfaces

All data output via data output interface is inhibited when the module is performing the pre-operational self-test, conditional self-tests, zeroization, or when the module enters an error state. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not implement authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 12: Roles

The module supports the Crypto Officer role only. This sole role is implicitly assumed by the operator of the module when performing a service.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Symmetric encryption	Perform AES encryption	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, Plaintext	Ciphertext	Encryption with AES	Crypto Officer - AES keys: W,E
Symmetric decryption	Perform AES decryption	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, Ciphertext	Plaintext	Decryption with AES	Crypto Officer - AES keys: W,E
Authenticated symmetric	Perform AES encryption and	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, Plaintext, IV	Ciphertext, MAC tag	Authenticated encryption with AES	Crypto Officer - AES keys: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
encryption	authentication					
Authenticated symmetric decryption	Perform AES decryption and authentication	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	AES key, Ciphertext, MAC tag	Plaintext or fail	Authenticated decryption with AES	Crypto Officer - AES keys: W,E
RSA Key generation	Generate RSA key pairs	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) returns GPG_ERR_NO_ERROR	Key size	RSA public key, RSA private key	Key Pair Generation with RSA	Crypto Officer - RSA public keys: G,R - RSA private keys: G,R
ECDSA Key generation	Generate ECDSA key pairs	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) returns GPG_ERR_NO_ERROR	Key size	ECDSA public key, ECDSA private key	Key Pair Generation with ECDSA	Crypto Officer - ECDSA public keys: G,R - ECDSA private keys: G,R
ECDSA Digital signature generation	ECDSA signature generation	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	ECDSA private key, message, hash algorithm	Signature	Signature Generation with ECDSA	Crypto Officer - ECDSA private keys: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
RSA Digital signature generation	RSA signature generation	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	RSA private key, message, hash algorithm	Signature	Signature Generation with RSA	Crypto Officer - RSA private keys: W,E
ECDSA Digital signature verification	ECDSA signature verification	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	Signature, hash algorithm, ECDSA public key	Signature verification result	Signature Verification with ECDSA	Crypto Officer - ECDSA public keys: W,E
Digital signature verification	RSA signature verification	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS, ...) and gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) return GPG_ERR_NO_ERROR	Signature, hash algorithm, RSA public key	Signature verification result	Signature Verification with RSA	Crypto Officer - RSA public keys: W,E
Public key verification	Verify ECDSA public key	gcry_mpi_ec_curve_point() returns GPG_ERR_NO_ERROR	ECDSA public key, ECDSA private key	Return codes/log messages	Public Key Verification with ECDSA	Crypto Officer - ECDSA public keys: W,E
Random number generation	Generate random bitstrings	gcry_randomize(), gcry_random_bytes(), gcry_random_bytes_secure() return GPG_ERR_NO_ERROR	Size	Random number	Random Number Generation with DRBG	Crypto Officer - Entropy input: W,E - DRBG seed: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DRBG internal state (V value, key): W,E - DRBG internal state (V value, C value): W,E
Message digest	Compute SHA hashes	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MD, ...) returns GPG_ERR_NO_ERROR	Message	Message digest	Hashes Extendable Output Functions	Crypto Officer
Message authentication code (MAC)	Compute HMAC or AES-based CMAC	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_MAC, ...) returns GPG_ERR_NO_ERROR	Message, key	MAC tag	Message Authentication Code	Crypto Officer - HMAC keys: W,E - AES keys: W,E
Key wrapping	Perform AES-based key wrapping	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	Key wrapping key, key to be wrapped	Wrapped key	Key wrapping using AES CCM Key wrapping using AES KW	Crypto Officer - AES keys: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key unwrapping	Perform AES-based unwrapping	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER, ...) returns GPG_ERR_NO_ERROR	Wrapped key, key unwrapping key	Unwrapped key	Key unwrapping using AES CCM Key unwrapping using AES KW	Crypto Officer - AES keys: W,E
Key derivation	Perform key derivation	gcry_control(GCRYCTL_FIPS_SERVICE_INDICATOR_KDF, ...) returns GPG_ERR_NO_ERROR	Password/passphrase; Derived key	Derived key	Key Derivation with PBKDF	Crypto Officer - Derived key: G,R - Password or passphrase: W,E
Show status	Show module status	N/A	None	Current status of the module	None	Crypto Officer
Zeroization	Zeroize SSPs	N/A	N/A	N/A	None	Crypto Officer - AES keys: Z - HMAC keys: Z - RSA public keys: Z - RSA private keys: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						ECDSA public keys: Z - ECDSA private keys: Z - Password or passphrase: Z - Derived key: Z - Entropy input: Z - DRBG internal state (V value, key): Z - DRBG internal state (V value, C value): Z - DRBG seed: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Self-tests	Perform self-tests	N/A	Power on of the module	N/A	None	Crypto Officer
Show module name and version	Show module name and version	N/A	N/A	Display module name and version	None	Crypto Officer

Table 13: Approved Services

The table above lists the approved services. For each service, the table lists the associated cryptographic algorithm(s), the role to perform the service, the cryptographic keys or CSPs involved, and their access type(s). The following convention is used to specify access rights to a CSP:

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

The details of the approved cryptographic algorithms including the CAVP certificate numbers can be found in the Approved Algorithms table in Section 2.5 Algorithms. In order to check whether it utilizes an approved security function or not, the operator is responsible to invoke the gcry_control() API along with dedicated controls in the form of API input parameters.

The module implements the following controls depending on the requested service:

1. GCRYCTL_FIPS_SERVICE_INDICATOR_CIPHER - For symmetric algorithms and the related modes.
2. GCRYCTL_FIPS_SERVICE_INDICATOR_KDF - For KDF operations.
3. GCRYCTL_FIPS_SERVICE_INDICATOR_PK_FLAGS - For asymmetric operations.[1]
4. GCRYCTL_FIPS_SERVICE_INDICATOR_MD - For digest operations.
5. GCRYCTL_FIPS_SERVICE_INDICATOR_MAC - For MAC operations.

[1] The list of public key flags allowed in approved mode of operation is described in Appendix A. Approved Public Key Flags.

In addition to that, for the below-mentioned services, the approved service indicator corresponds to the GPG_ERR_NO_ERROR returned from listed functions in the indicator column below. They don't use gcry_control() API:

1. *Random number generation* service: gcry_randomize(), gcry_random_bytes(), gcry_random_bytes_secure().
2. *Public key validation* service: gcry_mpi_ec_curve_point().

For all approved services, GPG_ERR_NO_ERROR (i.e., "0") return code indicates the service is approved. In case the above-mentioned controls are used in conjunction, the operator is responsible to check that all the called functions return GPG_ERR_NO_ERROR (i.e., "0"). For all non-approved services, "non-zero" return code indicates the service is not approved.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Symmetric encryption	AES encryption using non-approved AES modes	AES-GCM, AES-GCM-SIV, AES-OCB, AES-EAX	CO
Symmetric decryption	AES decryption using non-approved AES modes	AES-GCM, AES-GCM-SIV, AES-OCB, AES-EAX	CO
Message digest	Message digest using non-approved algorithms	MD5	CO
Shared Secret Computation	ECDH Shared Secret Computation	ECDH	CO
Key generation	Generate RSA/ECDSA key pairs with non-approved public key flags	RSA ECDSA	CO
Digital signature generation	RSA/ECDSA signature generation with non-approved public key flags	RSA with non-approved public key flags ECDSA with non-approved public key flags	CO
Digital signature verification	RSA/ECDSA signature verification with non-approved public key flags	RSA with non-approved public key flags ECDSA with non-approved public key flags	CO
Asymmetric encryption primitives	RSA encryption primitives	RSA	CO

Name	Description	Algorithms	Role
Asymmetric decryption primitives	RSA decryption primitives	RSA	CO
Signature generation primitives	RSA/ECDSA signature generation primitives	RSA ECDSA	CO
Signature verification primitives	RSA/ECDSA signature verification primitives	RSA ECDSA	CO

Table 14: Non-Approved Services

The table above lists the non-approved services. The details of the non-approved cryptographic algorithms not available in non-approved mode can be found in the Not Allowed, Non-Approved Algorithms table. For the services listed above, the module implements an additional service indicator in the form of a control named GCRYCTL_FIPS_SERVICE_INDICATOR_FUNCTION. The operator is responsible to invoke the gcry_control() API along with the following input parameters: GCRYCTL_FIPS_SERVICE_INDICATOR_FUNCTION control; the name of the API[2] representing the service.

4.5 External Software/Firmware Loaded

The module does not load any external software/firmware.

[2] The list of APIs supported by the module can be found in the documentation included in the optional *libgcrypt-devel* package.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing the HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the module's ELF header that was computed at build time for each software component of the module. If the HMAC values do not match, the test fails, and the module enters the Error state.

5.2 Initiate on Demand

Integrity test is performed as part of the Pre-Operational Self-Test.

The module provides the Self-Test service to perform self-tests on demand which includes the pre-operational self-test (i.e., integrity test) and cryptographic algorithm self-tests (CASTs). This service can be invoked relying on the gcry_control(GCRYCTL_SELFTEST) API function call or by powering-off and reloading the module. During the execution of the on-demand self-tests, services are not available, and no data output or input is possible.

In order to verify whether the self-tests have succeeded and the module is in the Operational state, the calling application may invoke the gcry_control(GCRYCTL_OPERATIONAL_P). The function will return TRUE if the module is in the Operational state and FALSE if the module is in the Error state.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

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

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11 Life-Cycle Assurance.

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

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution	Dynamic

Table 15: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form. SSPs are provided to the module by the calling process and are destroyed when released by the appropriate zeroization function calls.

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 16: SSP Input-Output Methods

The module does not support manual SSP input or intermediate SSP generation output. The SSPs are provided to the module via API input parameters in plaintext form and output via API output parameters in plaintext form within the physical perimeter of the operational environment. This is allowed by FIPS 140-3 IG 9.5.A, according to the “CM Software to/from App via TOEPP Path” entry in the table above.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the provided cipher handle	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable. The completion of a zeroization routine will indicate that a zeroization procedure succeeded.	By calling the appropriate zeroization functions: AES key: gcry_cipher_close HMAC key: gcry_mac_close, gcry_free Key-derivation key: gcry_free Derived key: gcry_free RSA keys: gcry_mpi_release, gcry_sexp_release, gcry_free EC keys: gcry_mpi_release, gcry_free, gcry_mpi_point_release, gcry_sexp_release, gcry_ctx_release Entropy input: gcry_ctrl(GCRYCTL_TERM_SECMEM) Internal state: gcry_ctrl(GCRYCTL_TERM_SECMEM)
Remove power from the module	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed. Module power off indicates that the zeroization procedure succeeded.	By unloading the module

Table 17: SSP Zeroization Methods

The memory occupied by SSPs is allocated by regular memory allocation operating system calls. The application that is acting as the CO is responsible for calling the appropriate zeroization functions provided in the module's API and listed in the table above. Calling gcry_free() will zeroize the SSPs and also invoke the corresponding API functions listed in table to zeroize SSPs. The zeroization functions overwrite the memory occupied by SSPs with “zeros” and deallocate the memory with the regular memory deallocation operating system call. In case of abnormal termination, or swap in/out of a physical memory page of a process, the keys in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process. Data output via the data output interface is inhibited during zeroization. The completion of a zeroization routine(s) will indicate that a zeroization procedure succeeded.

The user must not call malloc/free to create/release space for keys, and must let libgcrypt manage space for keys, which ensures that the key memory is overwritten before it is released. gcry_control(GCRYCTL_TERM_SECMEM) needs to be called before the process is terminated.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES keys	AES key used for encryption, decryption, key wrapping, key unwrapping, and computing MAC tags	XTS: 256, 512 bits; Other modes: 128, 192, 256 bits - XTS: 128, 256 bits; Other modes; 128, 192, 256 bits	Symmetric key - CSP			Key wrapping using AES CCM Key wrapping using AES KW Key unwrapping using AES CCM Key unwrapping using AES KW Encryption with AES Decryption with AES
HMAC keys	HMAC key used for message authentication code	112 to 256 bits - 112 to 256 bits	Symmetric key - CSP			Message Authentication Code
RSA public keys	Public key used for RSA signature verification	1024, 1536, 2048, 3072, 4096 bits - 80, 96, 112, 128, 149 bits	Public key - PSP	Key Pair Generation with RSA		Key Pair Generation with RSA Signature Verification with RSA
RSA private keys	Private key used for RSA signature generation	2048, 3072, 4096 bits - 112, 128, 149 bits	Private key - CSP	Key Pair Generation with RSA		Key Pair Generation with RSA Signature Generation with RSA
ECDSA public keys	Public key used for ECDSA signature verification	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation with ECDSA		Key Pair Generation with ECDSA Signature

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Verification with ECDSA
ECDSA private keys	Private key used for ECDSA signature generation	P-224, P-256, P-384, P-521 - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation with ECDSA		Key Pair Generation with ECDSA Signature Generation with ECDSA
Password or passphrase	PBKDF2 password	At least 8 characters - N/A	Password or passphrase - CSP			Key Derivation with PBKDF
Derived key	PBKDF2 derived key	112-256 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with PBKDF		Key Derivation with PBKDF
Entropy input	Entropy input used to seed the DRBGs	128-384 bits - 128-256 bits	Entropy input - CSP			Random Number Generation with DRBG
DRBG internal state (V value, key)	Internal state of CTR_DRBG and HMAC_DRBG	CTR_DRBG: 256, 320, 348 bits; HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
DRBG internal state (V value, C value)	Internal state of Hash_DRBG	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG seed	DRBG seed derived from entropy input	CTR_DRBG: 256, 320, 348 bits; Hash DRBG: 440, 888 bits; HMAC_DRBG: 160, 256, 512 bits - CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRBG: 128, 256 bits	Seed - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES keys	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	
HMAC keys	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	
RSA public keys	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	RSA private keys:Paired With
RSA private keys	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	RSA public keys:Paired With
ECDSA public keys	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	ECDSA private keys:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ECDSA private keys	API input parameters API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	ECDSA public keys:Paired With
Password or passphrase	API input parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Derived key:Derives
Derived key	API output parameters	RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Password or passphrase:Derived From
Entropy input		RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	DRBG seed:Generates
DRBG internal state (V value, key)		RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	DRBG seed:Generated from
DRBG internal state (V value, C value)		RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	DRBG seed:Generated from
DRBG seed		RAM:Plaintext	For the duration of the service	Free cipher handle Remove power from the module	Entropy input:Generated from DRBG internal state (V value, key):Generates DRBG internal state (V value, C value):Generates

Table 19: 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 RSA and ECDSA algorithms as implemented by the module conform to FIPS 186-4, which has been superseded by FIPS 186-5. FIPS 186-4 has been withdrawn since February 3, 2024.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A4597)	Key size: 376 bits	Message Authentication	SW/FW Integrity	Module is operational	Integrity test for libgcrypt.so.20.4.2
HMAC-SHA2-256 (A4598)	Key size: 376 bits	Message Authentication	SW/FW Integrity	Module is operational	Integrity test for libgcrypt.so.20.4.2
HMAC-SHA2-256 (A4600)	Key size: 376 bits	Message Authentication	SW/FW Integrity	Module is operational	Integrity test for libgcrypt.so.20.4.2
HMAC-SHA2-256 (A4601)	Key size: 376 bits	Message Authentication	SW/FW Integrity	Module is operational	Integrity test for libgcrypt.so.20.4.2
HMAC-SHA2-256 (A4602)	Key size: 376 bits	Message Authentication	SW/FW Integrity	Module is operational	Integrity test for libgcrypt.so.20.4.2

Table 20: Pre-Operational Self-Tests

The details of integrity test are provided in Section 5.1 Integrity Techniques. Data output via the data output interface is inhibited during the execution of the pre-operational self-test.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A4597)	AES ECB mode with 128, 192, 256-bit keys for encryption	KAT	CAST	Module is operational	Encryption	Module initialization or on demand through API function call
AES-ECB (A4598)	AES ECB mode with 128, 192, 256-bit keys for encryption	KAT	CAST	Module is operational	Encryption	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
AES-ECB (A4600)	AES ECB mode with 128, 192, 256-bit keys for encryption	KAT	CAST	Module is operational	Encryption	Module initialization or on demand through API function call
AES-ECB (A4601)	AES ECB mode with 128, 192, 256-bit keys for encryption	KAT	CAST	Module is operational	Encryption	Module initialization or on demand through API function call
AES-ECB (A4597)	AES ECB mode with 128, 192, 256-bit keys for decryption	KAT	CAST	Module is operational	Decryption	Module initialization or on demand through API function call
AES-ECB (A4598)	AES ECB mode with 128, 192, 256-bit keys for decryption	KAT	CAST	Module is operational	Decryption	Module initialization or on demand through API function call
AES-ECB (A4600)	AES ECB mode with 128, 192, 256-bit keys for decryption	KAT	CAST	Module is operational	Decryption	Module initialization or on demand through API function call
AES-ECB (A4601)	AES ECB mode with 128, 192, 256-bit keys for decryption	KAT	CAST	Module is operational	Decryption	Module initialization or on demand through API function call
AES-CMAC (A4597)	AES CMAC with 128-bit key, MAC generation	KAT	CAST	Module is operational	Message Authentication	Module initialization or

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						on demand through API function call
AES-CMAC (A4598)	AES CMAC with 128-bit key, MAC generation	KAT	CAST	Module is operational	Message Authentication	Module initialization or on demand through API function call
AES-CMAC (A4600)	AES CMAC with 128-bit key, MAC generation	KAT	CAST	Module is operational	Message Authentication	Module initialization or on demand through API function call
AES-CMAC (A4597)	AES CMAC with 128-bit key, MAC generation	KAT	CAST	Module is operational	Message Authentication	Module initialization or on demand through API function call
Counter DRBG (A4597)	CTR_DRBG with 129-bit key with DF, with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Counter DRBG (A4598)	CTR_DRBG with 129-bit key with DF, with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Counter DRBG (A4600)	CTR_DRBG with 129-bit key with DF, with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG (A4601)	CTR_DRBG with 129-bit key with DF, with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Hash DRBG (A4597)	SHA-256 with and without PR; SHA-1 without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Hash DRBG (A4598)	SHA-256 with and without PR; SHA-1 without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Hash DRBG (A4597)	SHA-256 with and without PR; SHA-1 without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Hash DRBG (A4597)	SHA-256 with and without PR; SHA-1 without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
Hash DRBG (A4597)	SHA-256 with and without PR; SHA-1 without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
HMAC DRBG (A4597)	SHA-256 with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
HMAC DRBG (A4598)	SHA-256 with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
HMAC DRBG (A4600)	SHA-256 with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
HMAC DRBG (A4601)	SHA-256 with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
HMAC DRBG (A4602)	SHA-256 with and without PR	KAT	CAST	Module is operational	Compliant with section 11.3 of SP 800-90Ar1	Module initialization or on demand through API function call
ECDSA SigGen (FIPS186-4) (A4597)	ECDSA signature generation with P-256 and SHA-256	KAT	CAST	Module is operational	Signature generation	Module initialization or on demand through API function call
ECDSA SigGen (FIPS186-4) (A4598)	ECDSA signature generation with P-256 and SHA-256	KAT	CAST	Module is operational	Signature generation	Module initialization or on demand through API function call
ECDSA SigGen	ECDSA signature generation with P-256 and SHA-256	KAT	CAST	Module is operational	Signature generation	Module initialization or

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A4600)						on demand through API function call
ECDSA SigGen (FIPS186-4) (A4601)	ECDSA signature generation with P-256 and SHA-256	KAT	CAST	Module is operational	Signature generation	Module initialization or on demand through API function call
ECDSA SigGen (FIPS186-4) (A4602)	ECDSA signature generation with P-256 and SHA-256	KAT	CAST	Module is operational	Signature generation	Module initialization or on demand through API function call
ECDSA SigVer (FIPS186-4) (A4597)	ECDSA signature verification with P-256 and SHA-256	KAT	CAST	Module is operational	Signature verification	Module initialization or on demand through API function call
ECDSA SigVer (FIPS186-4) (A4598)	ECDSA signature verification with P-256 and SHA-256	KAT	CAST	Module is operational	Signature verification	Module initialization or on demand through API function call
ECDSA SigVer (FIPS186-4) (A4600)	ECDSA signature verification with P-256 and SHA-256	KAT	CAST	Module is operational	Signature verification	Module initialization or on demand through API function call
ECDSA SigVer (FIPS186-4) (A4601)	ECDSA signature verification with P-256 and SHA-256	KAT	CAST	Module is operational	Signature verification	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigVer (FIPS186-4) (A4602)	ECDSA signature verification with P-256 and SHA-256	KAT	CAST	Module is operational	Signature verification	Module initialization or on demand through API function call
HMAC-SHA-1 (A4596)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA-1 (A4597)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA-1 (A4598)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA-1 (A4599)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA-1 (A4600)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA-1 (A4601)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
HMAC-SHA-1 (A4602)	HMAC-SHA-1	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-224 (A4597)	HMAC-SHA-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-224 (A4598)	HMAC-SHA-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-224 (A4600)	HMAC-SHA-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-224 (A4601)	HMAC-SHA-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-224 (A4602)	HMAC-SHA-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A4597)	HMAC-SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-256 (A4598)	HMAC-SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-256 (A4600)	HMAC-SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-256 (A4601)	HMAC-SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-256 (A4602)	HMAC-SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-384 (A4597)	HMAC-SHA-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-384 (A4598)	HMAC-SHA-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
HMAC-SHA2-384 (A4600)	HMAC-SHA-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-384 (A4601)	HMAC-SHA-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-384 (A4602)	HMAC-SHA-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-512 (A4597)	HMAC-SHA-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-512 (A4598)	HMAC-SHA-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-512 (A4600)	HMAC-SHA-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-512 (A4601)	HMAC-SHA-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA2-512 (A4602)	HMAC-SHA-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-224 (A4597)	HMAC-SHA3-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-224 (A4597)	HMAC-SHA3-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-224 (A4597)	HMAC-SHA3-224	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-256 (A4597)	HMAC-SHA3-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-256 (A4598)	HMAC-SHA3-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
HMAC-SHA3-256 (A4602)	HMAC-SHA3-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-384 (A4597)	HMAC-SHA3-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-384 (A4598)	HMAC-SHA3-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-384 (A4602)	HMAC-SHA3-384	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-512 (A4597)	HMAC-SHA3-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
HMAC-SHA3-512 (A4598)	HMAC-SHA3-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA3-512 (A4602)	HMAC-SHA3-512	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigGen (FIPS186-4) (A4597)	PKCS#1 v1.5 with 2048-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigGen (FIPS186-4) (A4598)	PKCS#1 v1.5 with 2048-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigGen (FIPS186-4) (A4597)	PKCS#1 v1.5 with 2048-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigGen (FIPS186-4) (A4601)	PKCS#1 v1.5 with 2048-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigGen (FIPS186-4) (A4602)	PKCS#1 v1.5 with 2048-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigVer (FIPS186-4) (A4597)	PKCS#1 v1.5 with 2084-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
RSA SigVer (FIPS186-4) (A4598)	PKCS#1 v1.5 with 2084-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigVer (FIPS186-4) (A4600)	PKCS#1 v1.5 with 2084-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigVer (FIPS186-4) (A4601)	PKCS#1 v1.5 with 2084-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
RSA SigVer (FIPS186-4) (A4602)	PKCS#1 v1.5 with 2084-bit key and SHA-256	KAT	CAST	Module is operational	Message authentication	Module initialization or on demand through API function call
SHA-1 (A4596)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA-1 (A4597)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA-1 (A4598)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						on demand through API function call
SHA-1 (A4600)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA-1 (A4601)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA-1 (A4602)	SHA-1	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-224 (A4597)	SHA-224	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-224 (A4598)	SHA-224	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-224 (A4600)	SHA-224	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-224 (A4601)	SHA-224	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-224 (A4602)	SHA-224	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-256 (A4597)	SHA-256	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-256 (A4598)	SHA-256	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-256 (A4600)	SHA-256	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-256 (A4601)	SHA-256	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-256 (A4602)	SHA-256	KAT	CAST	Module is operational	Message digest	Module initialization or on demand

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						through API function call
SHA2-384 (A4597)	SHA-384	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-384 (A4598)	SHA-384	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-384 (A4600)	SHA-384	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-384 (A4601)	SHA-384	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-384 (A4602)	SHA-384	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-512 (A4597)	SHA-512	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-512 (A4598)	SHA-512	KAT	CAST	Module is operational	Message digest	Module initialization or

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						on demand through API function call
SHA2-512 (A4600)	SHA-512	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-512 (A4601)	SHA-512	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
SHA2-512 (A4602)	SHA-512	KAT	CAST	Module is operational	Message digest	Module initialization or on demand through API function call
PBKDF (A4597)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits; SHA-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module is operational	Password-based key derivation	Module initialization or on demand through API function call
PBKDF (A4598)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits; SHA-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module is operational	Password-based key derivation	Module initialization or on demand through API function call

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
PBKDF (A4600)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits; SHA-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module is operational	Password-based key derivation	Module initialization or on demand through API function call
PBKDF (A4601)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits; SHA-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module is operational	Password-based key derivation	Module initialization or on demand through API function call
PBKDF (A4602)	SHA-1 password length 24 characters, master key length of 200 bits, iteration count of 4096, and salt length of 288 bits; SHA-256 password length 24 characters, master key length of 320 bits, iteration count of 4096, and salt length of 288 bits	KAT	CAST	Module is operational	Password-based key derivation	Module initialization or on demand through API function call
ECDSA KeyGen (FIPS186-4) (A4597)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	EC key pair generation	Key generation
ECDSA KeyGen (FIPS186-4) (A4598)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	EC key pair generation	Key generation
ECDSA KeyGen	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	EC key pair generation	Key generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A4600)						
ECDSA KeyGen (FIPS186-4) (A4601)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	EC key pair generation	Key generation
ECDSA KeyGen (FIPS186-4) (A4602)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	EC key pair generation	Key generation
RSA KeyGen (FIPS186-4) (A4597)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	RSA key pair generation	Key generation
RSA KeyGen (FIPS186-4) (A4598)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	RSA key pair generation	Key generation
RSA KeyGen (FIPS186-4) (A4600)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	RSA key pair generation	Key generation
RSA KeyGen (FIPS186-4) (A4601)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	RSA key pair generation	Key generation
RSA KeyGen (FIPS186-4) (A4602)	Signature generation and verification with SHA-256	PCT	PCT	Successful key generation	RSA key pair generation	Key generation

Table 21: Conditional Self-Tests

The CASTs are run prior to performing the integrity test. Data output via the data output interface is inhibited during the execution of the conditional self-tests.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4597)	Message Authentication	SW/FW Integrity	Whenever the module is powered on	Upon every power on
HMAC-SHA2-256 (A4598)	Message Authentication	SW/FW Integrity	Whenever the module is powered on	Upon every power on
HMAC-SHA2-256 (A4600)	Message Authentication	SW/FW Integrity	Whenever the module is powered on	Upon every power on
HMAC-SHA2-256 (A4601)	Message Authentication	SW/FW Integrity	Whenever the module is powered on	Upon every power on
HMAC-SHA2-256 (A4602)	Message Authentication	SW/FW Integrity	Whenever the module is powered on	Upon every power on

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A4597)	KAT	CAST	On demand	Manually
AES-ECB (A4598)	KAT	CAST	On demand	Manually
AES-ECB (A4600)	KAT	CAST	On demand	Manually
AES-ECB (A4601)	KAT	CAST	On demand	Manually
AES-ECB (A4597)	KAT	CAST	On demand	Manually
AES-ECB (A4598)	KAT	CAST	On demand	Manually
AES-ECB (A4600)	KAT	CAST	On demand	Manually
AES-ECB (A4601)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CMAC (A4597)	KAT	CAST	On demand	Manually
AES-CMAC (A4598)	KAT	CAST	On demand	Manually
AES-CMAC (A4600)	KAT	CAST	On demand	Manually
AES-CMAC (A4597)	KAT	CAST	On demand	Manually
Counter DRBG (A4597)	KAT	CAST	On demand	Manually
Counter DRBG (A4598)	KAT	CAST	On demand	Manually
Counter DRBG (A4600)	KAT	CAST	On demand	Manually
Counter DRBG (A4601)	KAT	CAST	On demand	Manually
Hash DRBG (A4597)	KAT	CAST	On demand	Manually
Hash DRBG (A4598)	KAT	CAST	On demand	Manually
Hash DRBG (A4597)	KAT	CAST	On demand	Manually
Hash DRBG (A4597)	KAT	CAST	On demand	Manually
Hash DRBG (A4597)	KAT	CAST	On demand	Manually
HMAC DRBG (A4597)	KAT	CAST	On demand	Manually
HMAC DRBG (A4598)	KAT	CAST	On demand	Manually
HMAC DRBG (A4600)	KAT	CAST	On demand	Manually
HMAC DRBG (A4601)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A4602)	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-4) (A4597)	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-4) (A4598)	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-4) (A4600)	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-4) (A4601)	KAT	CAST	On demand	Manually
ECDSA SigGen (FIPS186-4) (A4602)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-4) (A4597)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-4) (A4598)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-4) (A4600)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-4) (A4601)	KAT	CAST	On demand	Manually
ECDSA SigVer (FIPS186-4) (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4596)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4598)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA-1 (A4599)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4600)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4601)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A4600)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A4601)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4600)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4601)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A4602)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-384 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A4600)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A4601)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A4600)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A4601)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA3-224 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA3-224 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA3-224 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA3-256 (A4597)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA3-256 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA3-256 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA3-384 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA3-384 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA3-384 (A4602)	KAT	CAST	On demand	Manually
HMAC-SHA3-512 (A4597)	KAT	CAST	On demand	Manually
HMAC-SHA3-512 (A4598)	KAT	CAST	On demand	Manually
HMAC-SHA3-512 (A4602)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-4) (A4597)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-4) (A4598)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-4) (A4597)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-4) (A4601)	KAT	CAST	On demand	Manually
RSA SigGen (FIPS186-4) (A4602)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-4) (A4597)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A4598)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-4) (A4600)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-4) (A4601)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-4) (A4602)	KAT	CAST	On demand	Manually
SHA-1 (A4596)	KAT	CAST	On demand	Manually
SHA-1 (A4597)	KAT	CAST	On demand	Manually
SHA-1 (A4598)	KAT	CAST	On demand	Manually
SHA-1 (A4600)	KAT	CAST	On demand	Manually
SHA-1 (A4601)	KAT	CAST	On demand	Manually
SHA-1 (A4602)	KAT	CAST	On demand	Manually
SHA2-224 (A4597)	KAT	CAST	On demand	Manually
SHA2-224 (A4598)	KAT	CAST	On demand	Manually
SHA2-224 (A4600)	KAT	CAST	On demand	Manually
SHA2-224 (A4601)	KAT	CAST	On demand	Manually
SHA2-224 (A4602)	KAT	CAST	On demand	Manually
SHA2-256 (A4597)	KAT	CAST	On demand	Manually
SHA2-256 (A4598)	KAT	CAST	On demand	Manually
SHA2-256 (A4600)	KAT	CAST	On demand	Manually
SHA2-256 (A4601)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-256 (A4602)	KAT	CAST	On demand	Manually
SHA2-384 (A4597)	KAT	CAST	On demand	Manually
SHA2-384 (A4598)	KAT	CAST	On demand	Manually
SHA2-384 (A4600)	KAT	CAST	On demand	Manually
SHA2-384 (A4601)	KAT	CAST	On demand	Manually
SHA2-384 (A4602)	KAT	CAST	On demand	Manually
SHA2-512 (A4597)	KAT	CAST	On demand	Manually
SHA2-512 (A4598)	KAT	CAST	On demand	Manually
SHA2-512 (A4600)	KAT	CAST	On demand	Manually
SHA2-512 (A4601)	KAT	CAST	On demand	Manually
SHA2-512 (A4602)	KAT	CAST	On demand	Manually
PBKDF (A4597)	KAT	CAST	On demand	Manually
PBKDF (A4598)	KAT	CAST	On demand	Manually
PBKDF (A4600)	KAT	CAST	On demand	Manually
PBKDF (A4601)	KAT	CAST	On demand	Manually
PBKDF (A4602)	KAT	CAST	On demand	Manually
ECDSA KeyGen (FIPS186-4) (A4597)	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-4) (A4598)	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-4) (A4600)	PCT	PCT	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA KeyGen (FIPS186-4) (A4601)	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-4) (A4602)	PCT	PCT	On demand	Manually
RSA KeyGen (FIPS186-4) (A4597)	PCT	PCT	On demand	Manually
RSA KeyGen (FIPS186-4) (A4598)	PCT	PCT	On demand	Manually
RSA KeyGen (FIPS186-4) (A4600)	PCT	PCT	On demand	Manually
RSA KeyGen (FIPS186-4) (A4601)	PCT	PCT	On demand	Manually
RSA KeyGen (FIPS186-4) (A4602)	PCT	PCT	On demand	Manually

Table 23: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error state	The module immediately stops functioning due to a self-test failure	Software integrity test failure CAST failure PCT failure	Restart of the module	Module will not load; Module stops functioning for PCT failure
Fatal Error state	The module immediately halts all cryptographic operations and transits to shutdown	Random numbers are requested in the error state Cipher operations are requested on a deallocated handle	Restart of the module	Module is aborted and is not available for use

Table 24: Error States

The table above shows the error states and the corresponding condition. The calling application can obtain the module state by calling the gcry_control(GCRYCTL_OPERATIONAL_P) API function. The function returns FALSE if the module is in the Error state, TRUE if the module is in the Operational state.

When the module fails any pre-operational self-test or conditional self-tests, the module will return an error code to indicate the error and enter the Error state. If random numbers are requested in the Error state or cipher operations are requested on a deallocated handle, the module will enter the Fatal Error state. Any further cryptographic operation and all data output via the data output interface are inhibited in both error states. Recovering from the Error state includes performing self-tests and restarting the module. The only way to transition from the Fatal Error state to the Operational state is to restart the cryptographic module.

10.5 Operator Initiation of Self-Tests

The module provides the Self-Test service to perform self-tests as stated in Section 5.2 Initiate on Demand. During the execution of the on-demand self-tests, services are not available, and data output is not possible. Additionally, the PCTs can be invoked on demand by requesting the asymmetric key generation services.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Crypto Officer can install the RPM package of the module as listed in Section 11.2 Administrator Guidance using standard tools recommended for the installation of RPM packages on an Amazon Linux 2023 or SnowOS 1.0 system (for example, dnf, rpm, and the RHN remote management tool). The integrity of the RPM package is automatically verified during the installation, and the Crypto Officer shall not install the RPM package if there is any integrity error.

Before the RPM package of the module is installed, the Amazon Linux 2023 and SnowOS 1.0 systems must operate in the FIPS validated configuration. This can be achieved by executing the fips-mode-setup --enable command and then restarting the system. More information can be found at the vendor documentation.

The Crypto Officer must verify the Amazon Linux 2023 and SnowOS 1.0 system operates in the FIPS validated configuration by executing the fips-mode-setup --check command, which should output "FIPS mode is enabled."

11.2 Administrator Guidance

The binaries of the module are contained in the RPM packages for delivery. The Crypto Officer shall follow Section 11.1 Installation, Initialization, and Startup Procedures to configure the operational environment and install the module to be operated as a FIPS 140-3 validated module.

The following RPM packages contain the FIPS validated module:

- libgcrypt-1.10.2-1.amzn2023.0.2.x86_64.rpm for Intel 64-bit
- libgcrypt-1.10.2-1.amzn2023.0.2.aarch64.rpm for ARM 64-bit

After installation of the RPM package of the module, the operator must check the output of the gcry_get_config() API, which should include the following name and version:

Amazon Linux 2023 libgcrypt 1.10.2-e4ddf5ed7abe8d45

Once libgcrypt has been put into the FIPS validated configuration, it is not possible to switch back to standard mode without terminating the process first. If the logging verbosity level of libgcrypt has been set to at least 2, the state transitions and self-tests are logged.

The approved and non-approved security functions offered by the module are listed in Table 8, *Security Function Implementations*. The list of interfaces is provided in section 3. The Users responsibility on approved mode of operation is mentioned in section 2.4 that include description on mode of operation, section 2.7 that has algorithm specific information and in section 4.3 that lists approved services as well as service indicator mechanism. The module does not implement authentication methods.

11.3 Non-Administrator Guidance

There is no non-administrator guidance. The administrator guidance is specified in section 11.2

11.4 Design and Rules

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the libgcrypt-1.10.2-1.amzn2023.0.2 RPM package can be uninstalled from the Amazon Linux 2023 and SnowOS 1.0 systems.

12 Mitigation of Other Attacks

12.1 Attack List

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.

12.2 Mitigation Effectiveness

The module implements blinding against RSA Timing Attacks.

By default, the module uses the following blinding technique: instead of using the RSA decryption directly, a blinded value $y = x\,r^{e} \bmod n$ is decrypted and the unblinded value $x' = y'\,r^{-1} \bmod n$ returned.

The blinding value r is a random value with the size of the modulus n.

12.3 Guidance and Constraints

Not applicable.

12.4 Additional Information

Not applicable.

Appendix A. Approved Public Key Flags

Below are listed the approved public key flags for an input s-expression:

curve	d	data	e	ecdsa	flags	sig-val
genkey	hash	n	nbits	pkcs1	private-key	value
pss	public-key	q	r	raw	rsa	salt-length
rsa-use-e	s					

Appendix B. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
CAVP	Cryptographic Algorithm Validation Program
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
DF	Derivation Function
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAT	Known Answer Test
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PR	Prediction Resistance
PSP	Public Security Parameter
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm

SSP	Sensitive Security Parameter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix C. 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** January 2024 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf
FIPS180-4	**Secure Hash Standard (SHS)** March 2012 https://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 https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
SP800-38A	**NIST Special Publication 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
SP800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 https://csrc.nist.gov/publications/detail/sp/800-38b/final

SP800-38C	**NIST Special Publication 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
SP800-38E	**NIST Special Publication 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
SP800-38F	**NIST Special Publication 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
SP800-90Arev1	**NIST Special Publication 800-90A – Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP800-90B	**NIST Special Publication 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
SP800-132	**NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 https://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf
SP800-133rev2	**NIST Special Publication 800-133 – Revision 2 - 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 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf
SP800-140Br1	**NIST Special Publication 800-140B – Revision 1 - CMVP Security Policy Requirements** November 2023 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf