# Amazon Web Services, Inc.

## Amazon Linux 2023 Kernel Cryptographic API

# FIPS 140-3 Non-Proprietary Security Policy

**Document Version: 1.0**

**Document Date: 2026-06-10**

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 Amazon Linux 2023 Kernel Cryptographic API versions:

- **Kernel:** 6.1.41-64.118.amzn2023 (Amazon Linux 2023 on EC2 c7g.metal and EC2 c6i.metal) and 6.1.41-64.118.fips.amzn2023 (Amazon Linux 2023 on AWS Snowball, AWS Snowblade, AWS Snowcone)
- **Libkcapi:** 1.4.0-105.amzn2023

It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

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

## 1.2 Security Levels

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

| 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 | N/A |
| | Overall Level | 1 |

Table 1: Security Levels

## 1.3 Additional Information

This Security Policy describes the features and design of the module named Amazon Linux 2023 Kernel Cryptographic API using the terminology contained in the FIPS 140-3 specification. The FIPS 140-3 Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-3. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

This FIPS 140-3 non-proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

In preparing the Security Policy document, the laboratory formatted the vendor-supplied documentation for consolidation without altering the technical statements therein contained. The further refining of the Security

Policy document was conducted iteratively throughout the conformance testing, wherein the Security Policy was submitted to the vendor, who would then edit, modify, and add technical contents. The vendor would also supply additional documentation, which the laboratory formatted into the existing Security Policy, and resubmitted to the vendor for their final editing.

# 2 Cryptographic Module Specification

## 2.1 Description

**Purpose and Use:** The Amazon Linux 2023 Kernel Cryptographic API (hereafter referred to as "the module") provides a C language application program interface (API) for use by other (kernel space and user space) processes that require cryptographic functionality. The module operates on a general-purpose computer as part of the Linux kernel. Its cryptographic functionality can be accessed using the Linux Kernel Crypto API.

**Module Type**: Software

**Module Embodiment**: Multi-Chip Standalone

**Cryptographic Boundary:** The cryptographic boundary of the module is defined as the kernel binary and the kernel crypto object files, the libkcapi library, and the sha512hmac binary, which is used to verify the integrity of the software components. In addition, the cryptographic boundary contains the .hmac files which store the expected integrity values for each of the software components. The cryptographic boundary is denoted by the bold black line in Figure 1.

**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. The TOEPP is denoted by the thin grey line in Figure 1.

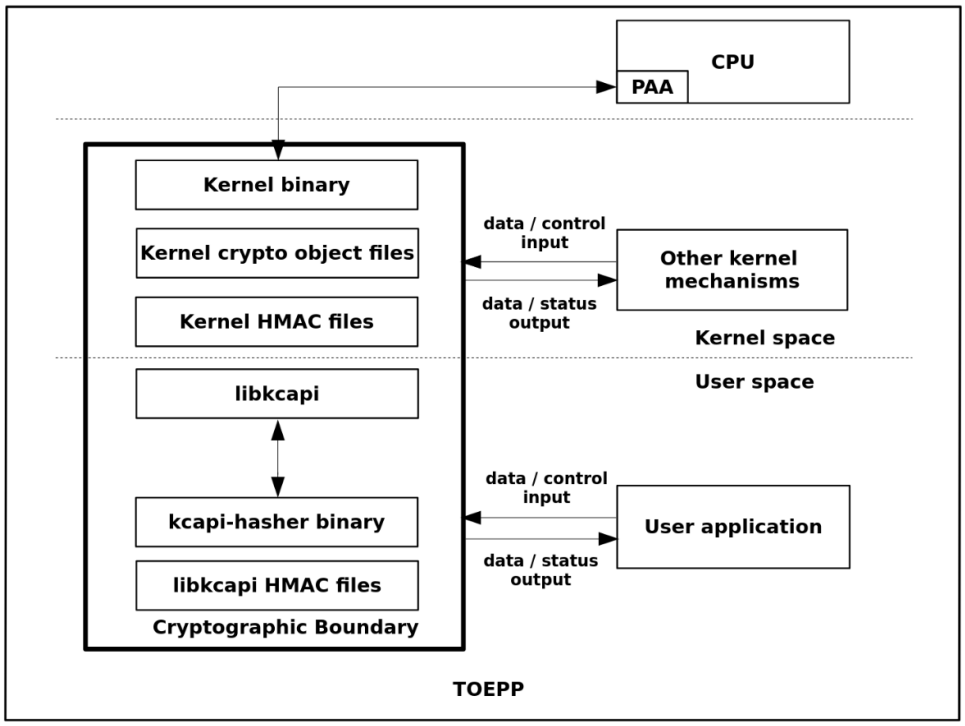


*Figure 1: Block Diagram*

## 2.2 Tested and Vendor Affirmed Module Version and Identification

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

| Package or File Name | Software/ Firmware Version | Features | Integrity Test |
|---|---|---|---|
| /boot/vmlinuz-6.1.41-64.118.fips.amzn2023.x86_64; *.ko and *.ko.xz files in /usr/lib/modules/6.1.41-64.118.fips.amzn2023.x86_64/kernel/crypto/ *.ko.xz files in /usr/lib/modules/6.1.41-64.118.fips.amzn2023.x86_64/kernel/arch/x86/crypto; /usr/lib64/libkcapi.so.1.4.0, /usr/lib/sha512hmac | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 | | HMAC-SHA-512 (vmlinuz, libkcapi.so.1.4.0, sha512hmac), RSA Signature Verification (*.ko files) |
| /boot/vmlinuz-6.1.41-64.118.amzn2023.aarch64; *.ko and *.ko.xz files in /usr/lib/modules/6.1.41-64.118.amzn2023.aarch64/kernel/crypto/, *.ko.xz files in /usr/lib/modules/6.1.41-64.118.amzn2023.aarch64/kernel/arch/aarch64/crypto; /usr/lib64/libkcapi.so.1.4.0, /usr/lib/sha512hmac | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 | | HMAC-SHA-512 (vmlinuz, libkcapi.so.1.4.0, sha512hmac), RSA Signature Verification (*.ko files) |
| /boot/vmlinuz-6.1.41-64.118.amzn2023.x86_64; *.ko and *.ko.xz files in /usr/lib/modules/6.1.41-64.118.amzn2023.x86_64/kernel/crypto/, *.ko.xz files in /usr/lib/modules/6.1.41-64.118.amzn2023.x86_64/kernel/arch/x86/crypto/; /usr/lib64/libkcapi.so.1.4.0, /usr/lib/sha512hmac | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 | | HMAC-SHA-512 (vmlinuz, libkcapi.so.1.4.0, sha512hmac), RSA Signature Verification (*.ko files) |

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

**Tested Operational Environments - Software, Firmware, Hybrid:**

| Operating System | Hardware Platform | Processors | PAA/PAI | Hypervisor or Host OS | Version(s) |
|---|---|---|---|---|---|
| Amazon Linux 2023 | EC2 c7g.metal | AWS Graviton3 | Yes | N/A | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | EC2 c6i.metal | Intel Xeon Platinum 8375C | Yes | N/A | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | AWS Snowball | AMD EPYC 7702 | Yes | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | AWS Snowblade | Intel Xeon Gold 6314U | Yes | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | AWS Snowcone | Intel Atom C3558 | Yes | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | EC2 c7g.metal | AWS Graviton3 | No | N/A | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | EC2 c6i.metal | Intel Xeon Platinum 8375C | No | N/A | Kernel: 6.1.41-64.118.amzn2023, libkcapi: 1.4.0-105.amzn2023 |

| Operating System | Hardware Platform | Processors | PAA/PAI | Hypervisor or Host OS | Version(s) |
|---|---|---|---|---|---|
| Amazon Linux 2023 | AWS Snowball | AMD EPYC 7702 | No | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | AWS Snowblade | Intel Xeon Gold 6314U | No | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |
| Amazon Linux 2023 | AWS Snowcone | Intel Atom C3558 | No | N/A | Kernel: 6.1.41-64.118.fips.amzn2023, libkcapi: 1.4.0-105.amzn2023 |

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

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

| Operating System | Hardware Platform |
|---|---|
| Bottlerocket v1.20.0 | EC2 c7g.metal with Intel Xeon Platinum 8375C (PAA: AES-NI) |
| Bottlerocket v1.20.0 | EC2 c6i.metal with AWS Graviton3 processor (PAA: Neon, Cryptography Extensions) |

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

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

## 2.3 Excluded Components

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

## 2.4 Modes of Operation

**Modes List and Description:**

| Mode Name | Description | Type | Status Indicator |
|---|---|---|---|
| Approved mode | Automatically entered whenever an approved service is requested | Approved | Mapped to approved service indicator in Section 4.3 for all approved algorithms except GCM: respective approved service function returns indicator 0. For GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set |
| Non-approved mode | Automatically entered whenever a non-approved service is requested | Non-Approved | No service indicator required for non-approved services per IG 2.4.C |

Table 5: Modes List and Description

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

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

## 2.5 Algorithms

**Approved Algorithms:**

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| AES-CBC | A4551, A4554, A4557, A4558, A4561, A4563, A4566 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |
| AES-CBC-CS3 | A4551, A4554, A4557, A4558, A4566 | Direction - decrypt, encrypt<br>Key Length - 128, 192, 256<br>Payload Length - Payload Length: 128-65536 Increment 8 | SP 800-38A |
| AES-CCM | A4551, A4554, A4557, A4558, A4566 | Key Length - 128, 192, 256<br>Tag Length - 112, 128, 64, 80, 96<br>IV Length - IV Length: 56, 64, 72, 80, 88, 96, 104<br>Payload Length - Payload Length: 0-256 Increment 8<br>AAD Length - AAD Length: 0, 256, 65536 | SP 800-38C |
| AES-CFB128 | A4551, A4554, A4558, A4566 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |
| AES-CMAC | A4551, A4554, A4557, A4558, A4566 | Direction - Generation, Verification<br>Key Length - 128, 192, 256<br>MAC Length - MAC Length: 128<br>Message Length - Message Length: 8-524288 Increment 8 | SP 800-38B |
| AES-CTR | A4551, A4554, A4557, A4558, A4561, A4563, A4566 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256<br>Payload Length - Payload Length: 8-128 Increment 8<br>Supports Counter larger than maximum value - No<br>Incremental Counter - Yes<br>Counter Tests Performed - Yes | SP 800-38A |
| AES-ECB | A4551, A4552, A4553, A4554, A4555, A4556, A4557, A4558, A4559, A4560, A4561, A4563, A4564, A4565, A4566, A4567, A4568 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |
| AES-GCM | A4551, A4554, A4558, A4566 | Direction - Decrypt, Encrypt<br>IV Generation - External<br>IV Generation Mode - 8.2.2<br>Key Length - 128, 192, 256<br>Tag Length - 104, 112, 120, 128, 32, 64, 96<br>IV Length - IV Length: 96<br>Payload Length - Payload Length: 128-65536 Increment 128<br>AAD Length - AAD Length: 128, 256, 120, 0 | SP 800-38D |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| AES-GCM | A4552, A4555, A4559, A4564, A4567 | Direction - Encrypt<br>IV Generation - Internal<br>IV Generation Mode - 8.2.2<br>Key Length - 128, 192, 256<br>Tag Length - 128, 64, 96<br>IV Length - IV Length: 96<br>Payload Length - Payload Length: 128, 256, 120, 248<br>AAD Length - AAD Length: 64, 96 | SP 800-38D |
| AES-GCM | A4553, A4556, A4560, A4563, A4565, A4568 | Direction - Decrypt, Encrypt<br>IV Generation - External<br>IV Generation Mode - 8.2.2<br>Key Length - 128, 192, 256<br>Tag Length - 128, 64, 96<br>IV Length - IV Length: 96<br>Payload Length - Payload Length: 128, 256, 120, 248<br>AAD Length - AAD Length: 64, 96 | SP 800-38D |
| AES-GMAC | A4551, A4554, A4558, A4566 | Direction - Decrypt, Encrypt<br>IV Generation - External<br>IV Generation Mode - 8.2.2<br>Key Length - 128, 192, 256<br>Tag Length - 104, 112, 120, 128, 32, 64, 96<br>IV Length - IV Length: 96<br>AAD Length - AAD Length: 128, 256, 120, 0 | SP 800-38D |
| AES-KW | A4551, A4554, A4558, A4566 | Direction - Decrypt, Encrypt<br>Cipher - Cipher<br>Key Length - 128, 192, 256<br>Payload Length - Payload Length: 128-4096 Increment 128 | SP 800-38F |
| AES-OFB | A4551, A4554, A4558, A4566 | Direction - Decrypt, Encrypt<br>Key Length - 128, 192, 256 | SP 800-38A |
| AES-XTS Testing Revision 2.0 | A4551, A4554, A4557, A4558, A4561, A4563, A4566 | Direction - Decrypt, Encrypt<br>Key Length - 128, 256<br>Payload Length - Payload Length: 128-65536 Increment 128<br>Tweak Mode - Hex<br>Data Unit Length Matches Payload Length - Yes | SP 800-38E |
| Counter DRBG | A4551, A4552, A4553, A4554, A4555, A4556, A4558, A4559, A4560, A4563, A4564, A4565, A4566, A4567, A4568 | Prediction Resistance - No, Yes<br>Supports Reseed - Yes<br>Mode - AES-128, AES-192, AES-256<br>Derivation Function Enabled - Yes<br>Additional Input - Additional Input: 0<br>Entropy Input - Entropy Input: 128, | SP 800-90A Rev. 1 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| | | Entropy Input: 192, Entropy Input: 256<br>Nonce - Nonce: 128, Nonce: 64<br>Personalization String Length - Personalization String Length: 0<br>Returned Bits - 1024, 4096, 512 | |
| ECDSA KeyGen (FIPS186-4) | A4551 | Curve - P-256, P-384<br>Secret Generation Mode - Testing Candidates | FIPS 186-4 |
| Hash DRBG | A4551, A4569, A4570, A4571 | Prediction Resistance - No, Yes<br>Supports Reseed - No<br>Mode - SHA-1, SHA2-256, SHA2-512<br>Entropy Input - Entropy Input: 160, Entropy Input: 256<br>Nonce - Nonce: 160, Nonce: 256<br>Personalization String Length - Personalization String Length: 0<br>Additional Input - Additional Input: 0, 160, Additional Input: 0, 256<br>Returned Bits - 1024, 2048, 320 | SP 800-90A Rev. 1 |
| HMAC DRBG | A4551, A4569, A4570, A4571 | Prediction Resistance - No, Yes<br>Supports Reseed - No<br>Mode - SHA-1, SHA2-256, SHA2-512<br>Entropy Input - Entropy Input: 160, Entropy Input: 256<br>Nonce - Nonce: 160, Nonce: 256<br>Personalization String Length - Personalization String Length: 0<br>Additional Input - Additional Input: 0, 160, Additional Input: 0, 256<br>Returned Bits - 1024, 2048, 320 | SP 800-90A Rev. 1 |
| HMAC-SHA-1 | A4551, A4557, A4569, A4570, A4571 | MAC - MAC: 160<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-224 | A4551, A4557, A4561, A4562, A4569, A4570, A4571 | MAC - MAC: 224<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-256 | A4551, A4557, A4561, A4562, A4569, A4570, A4571 | MAC - MAC: 256<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-384 | A4551, A4557, A4562, A4569, A4570, A4571 | MAC - MAC: 384<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA2-512 | A4551, A4557, A4562, A4569, A4570, A4571 | MAC - MAC: 512<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| HMAC-SHA3-224 | A4551, A4557 | MAC - MAC: 224<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-256 | A4551, A4557 | MAC - MAC: 256<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-384 | A4551, A4557 | MAC - MAC: 384<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| HMAC-SHA3-512 | A4551, A4557 | MAC - MAC: 512<br>Key Length - Key Length: 112-524288 Increment 8 | FIPS 198-1 |
| KAS-ECC-SSC Sp800-56Ar3 | A4551 | Domain Parameter Generation Methods - P-256, P-384<br>Scheme -<br>ephemeralUnified -<br>KAS Role - initiator, responder | SP 800-56A Rev. 3 |
| KAS-FFC-SSC Sp800-56Ar3 | A4551 | Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192<br>Scheme -<br>dhEphem -<br>KAS Role - initiator, responder | SP 800-56A Rev. 3 |
| RSA SigVer (FIPS186-4) | A4551, A4569, A4570, A4571 | Signature Type - PKCS 1.5<br>Modulo - 2048, 3072, 4096<br>Hash Pair -<br>Hash Algorithm - SHA-1<br>Public Exponent Mode - Random | FIPS 186-4 |
| Safe Primes Key Generation | A4551 | Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 | SP 800-56A Rev. 3 |
| SHA-1 | A4551, A4557, A4569, A4570, A4571 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 180-4 |
| SHA2-224 | A4551, A4557, A4561, A4562, A4569, A4570, A4571 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 180-4 |
| SHA2-256 | A4551, A4557, A4561, A4562, A4569, A4570, A4571 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 180-4 |
| SHA2-384 | A4551, A4557, A4562, A4569, A4570, A4571 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 180-4 |
| SHA2-512 | A4551, A4557, A4562, A4569, A4570, A4571 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 180-4 |

| Algorithm | CAVP Cert | Properties | Reference |
|---|---|---|---|
| SHA3-224 | A4551, A4557 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 202 |
| SHA3-256 | A4551, A4557 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 202 |
| SHA3-384 | A4551, A4557 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 202 |
| SHA3-512 | A4551, A4557 | Message Length - Message Length: 0-65536 Increment 8<br>Large Message Sizes - 1, 2 | FIPS 202 |

Table 6: Approved Algorithms

The Approved Algorithms table lists all approved cryptographic algorithms of the module, including specific key lengths employed for approved services and implemented modes or methods of operation of the algorithms.

**Vendor-Affirmed Algorithms:**

| Name | Properties | Implementation | Reference |
|---|---|---|---|
| Cryptographic Key Generation (CKG) | Key Type:Asymmetric | N/A | SP 800-133r2 Section 4 example 1 |

Table 7: Vendor-Affirmed Algorithms

**Non-Approved, Allowed Algorithms:**

N/A for this module.

**Non-Approved, Allowed Algorithms with No Security Claimed:**

N/A for this module.

**Non-Approved, Not Allowed Algorithms:**

| Name | Use and Function |
|---|---|
| AES GCM with external IV | Encryption (not compliant to SP 800-56Br2) |
| KBKDF (libkcapi) | Key Derivation with implementation not tested by CAVP |
| HKDF (libkcapi) | Key Derivation with implementation not tested by CAVP |
| PBKDF2 (libkcapi) | Password-Based Key Derivation with implementation not tested by CAVP |
| RSA | Encryption Primitive; Decryption Primitive (not compliant to SP 800-56Br2) |
| RSA with PKCS#1 v1.5 padding | Signature Generation (pre-hashed message); Signature Verification (pre-hashed message); Key Encapsulation (not compliant to SP 800-56Br2); Key Un-encapsulation (not compliant to SP 800-56Br2) |

Table 8: Non-Approved, Not Allowed Algorithms

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

## 2.6 Security Function Implementations

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Encryption, Decryption | BC-UnAuth | Encryption, Decryption | | AES-CBC-CS3: (A4554, A4551, A4566, A4557, A4558)<br>AES-CFB128: (A4554, A4551, A4566, A4558)<br>AES-CTR: (A4563, A4554, A4551, A4566, A4557, A4558, A4561)<br>AES-ECB: (A4563, A4567, A4559, A4554, A4551, A4564, A4566, A4557, A4568, A4558, A4556, A4560, A4561, A4552, A4555, A4553, A4565)<br>AES-KW: (A4554, A4551, A4566, A4558)<br>AES-OFB: (A4554, A4551, A4566, A4558)<br>AES-XTS Testing Revision 2.0: (A4563, A4554, A4551, A4566, A4557, A4558, A4561)<br>AES-CBC: (A4563, A4554, A4551, A4566, A4557, A4558, A4561) |
| Authenticated Encryption, Authenticated Decryption | BC-Auth | Authenticated Encryption, Authenticated Decryption | | AES-GCM: (A4563, A4567, A4559, A4554, A4551, A4564, A4566, A4568, A4558, A4556, A4560, A4552, A4555, A4553, A4565)<br>AES-CCM: (A4554, A4551, A4566, A4557, A4558) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Message Authentication | MAC | Message Authentication | | AES-CMAC: (A4554, A4551, A4566, A4557, A4558)<br>AES-GMAC: (A4554, A4551, A4566, A4558)<br>HMAC-SHA-1: (A4569, A4551, A4571, A4557, A4570)<br>HMAC-SHA2-224: (A4569, A4551, A4571, A4557, A4562, A4570, A4561)<br>HMAC-SHA2-256: (A4569, A4551, A4571, A4557, A4562, A4570, A4561)<br>HMAC-SHA2-384: (A4569, A4551, A4571, A4557, A4562, A4570)<br>HMAC-SHA2-512: (A4569, A4551, A4571, A4557, A4562, A4570)<br>HMAC-SHA3-224: (A4551, A4557)<br>HMAC-SHA3-256: (A4551, A4557)<br>HMAC-SHA3-384: (A4551, A4557)<br>HMAC-SHA3-512: (A4551, A4557) |
| Random Number Generation | DRBG | Random Number Generation | | Counter DRBG: (A4563, A4567, A4559, A4554, A4551, A4564, A4566, A4568, A4558, A4556, A4560, A4552, A4555, A4553, A4565)<br>HMAC DRBG: (A4569, A4551, |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| | | | | A4571, A4570)<br>Hash DRBG: (A4569, A4551, A4571, A4570) |
| Key Pair Generation | AsymKeyPair-KeyGen<br>CKG | Key Pair Generation | | ECDSA KeyGen (FIPS186-4): (A4551)<br>Cryptographic Key Generation (CKG): () |
| Shared Secret Computation | KAS-SSC | Shared Secret Computation | | KAS-ECC-SSC Sp800-56Ar3: (A4551)<br>KAS-FFC-SSC Sp800-56Ar3: (A4551) |
| Digital Signature Verification | DigSig-SigVer | Digital Signature Verification | | RSA SigVer (FIPS186-4): (A4569, A4551, A4571, A4570) |
| Message Digest | SHA | Message Digest | | SHA-1: (A4569, A4551, A4571, A4557, A4570)<br>SHA2-224: (A4569, A4551, A4571, A4557, A4562, A4570, A4561)<br>SHA2-256: (A4569, A4551, A4571, A4557, A4562, A4570, A4561)<br>SHA2-384: (A4569, A4551, A4571, A4557, A4562, A4570)<br>SHA2-512: (A4569, A4551, A4571, A4557, A4562, A4570)<br>SHA3-224: (A4551, A4557)<br>SHA3-256: (A4551, A4557)<br>SHA3-384: (A4551, A4557)<br>SHA3-512: (A4551, A4557) |

| Name | Type | Description | Properties | Algorithms |
|---|---|---|---|---|
| Safe Primes Key Generation Function | AsymKeyPair-KeyGen<br>CKG | Safe Primes Key Generation Function | | Safe Primes Key Generation: (A4551)<br>Cryptographic Key Generation (CKG): () |

Table 9: Security Function Implementations

## 2.7 Algorithm Specific Information

### 2.7.1 AES GCM IV

The Crypto Officer shall consider the following requirements and restrictions when using the module.

For IPsec, the module offers the AES GCM implementation and uses the context of Scenario 1 of FIPS 140-3 IG C.H. The mechanism for IV generation is compliant with RFC 4106. IVs generated using this mechanism may only be used in the context of AES GCM encryption within the IPsec protocol.

The module does not implement IPsec. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. This application must use RFC 7296 compliant IKEv2 to establish the shared secret SKEYSEED from which the AES GCM encryption keys are derived.

The design of the IPsec protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

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

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. This service can be requested by invoking the crypto_aead_encrypt API function with an AES GCM handle. When this is the case, the API will not set an approved service indicator.

### 2.7.2 AES XTS

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

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

In accordance with IG C.I, the module implements a check to ensure that the two AES keys used in the AES-XTS algorithm are not identical. As the module does not implement symmetric key generation, this check is performed when the keys are input by the operator. AES-XTS keys (i.e., Key_1 and Key_2) entered into the module shall be generated and/or established independently according to NIST SP 800-133rev2, Section 6.3. for an approved use of AES-XTS.

### 2.7.3 Authenticated Encryption/Decryption

The module does not establish SSPs using an approved key transport scheme (KTS). However, it does offer approved algorithms that can be used by an external operator/application as part of an approved KTS.

### 2.7.4 KAS-SSC

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer approved algorithms that can be used by an external operator/application as part of an approved KAS.

### 2.7.5 RSA

For RSA signature verification, all supported, approved modulus sizes have been CAVP tested.

### 2.7.6 SP 800-56Ar3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the operator must use the Diffie-Hellman and elliptic curve Diffie-Hellman shared secret computation algorithms with the NVMe and Bluetooth related protocols. Additionally, the module's approved key pair generation service must be used to generate ephemeral Diffie-Hellman or EC Diffie-Hellman key pairs, or the key pairs must be obtained from another FIPS-validated module. As part of this service, the module will internally perform the full public key validation of the generated public key.

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

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer approved authenticated algorithms that can be used by an external operator/application as part of an approved KAS.

### 2.7.7 Legacy Use

Digital signature verification using SHA-1 is allowed for legacy use only. Digital signature generation using SHA-1 is non-approved and not allowed in approved services.

## 2.8 RBG and Entropy

| Cert Number | Vendor Name |
|---|---|
| E105 | Amazon Web Services |

Table 10: Entropy Certificates

| Name | Type | Operational Environment | Sample Size | Entropy per Sample | Conditioning Component |
|---|---|---|---|---|---|
| Amazon Kernel CPU Time Jitter RNG Entropy Source | Non-Physical | See Tested Operational Environment Table | 256 bits | Full entropy | SHA-256 (A4551) |

Table 11: Entropy Sources

The module implements three different Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90Ar1: CTR_DRBG, Hash_DRBG, and HMAC_DRBG. Each of these DRBG implementations can be instantiated by the operator of the module. When instantiated, these DRBGs can be used to generate random numbers for external usage.

Additionally, the module employs a specific HMAC-SHA-512 DRBG implementation for internal purposes (e.g. to generate asymmetric key pairs). The HMAC-SHA-512 DRBG is instantiated with a 384-bit entropy input (corresponding to 384 bits of entropy) and reseeded with a 256-bits long entropy input (corresponding to 256 bits of entropy). To obtain the 384-bit entropy input, outputs of multiple GetEntropy() calls are concatenated to

receive the entropy input length greater than 256 bits. The output is truncated to get the entropy input string which is not a multiple of 256.

The module complies with the Public Use Document for ESV certificate E105 seeding the aforementioned DRBG using the jent_kcapi_random function, which corresponds to the GetEntropy() function. The operational environment of the module is identical to the one listed on the ESV certificate. There are no maintenance requirements for the entropy source.

## 2.9 Key Generation

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

## 2.10 Key Establishment

The module implements shared secret computation methods as listed in the Security Function Implementations table.

## 2.11 Industry Protocols

AES-GCM with internal IV generation in the approved mode is compliant with RFC 4106 and shall only be used in conjunction with the IPsec protocol.

Diffie-Hellman and EC Diffie-Hellman shall only be used with the NVMe and Bluetooth related protocols.

No other parts of the NVMe, Bluetooth, or IPSec protocols, other than those mentioned above, have been tested by the CAVP and CMVP.

# 3 Cryptographic Module Interfaces

## 3.1 Ports and Interfaces

| Physical Port | Logical Interface(s) | Data That Passes |
|---|---|---|
| N/A | Data Input | API data input parameters, AF_ALG type sockets |
| N/A | Data Output | API output parameters, AF_ALG type sockets |
| N/A | Control Input | API function calls, API control input parameters, AF_ALG type sockets, kernel command line |
| N/A | Status Output | API return values, AF_ALG type sockets, kernel logs |

Table 12: Ports and Interfaces

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

# 4 Roles, Services, and Authentication

## 4.1 Authentication Methods

The module does not implement authentication.

## 4.2 Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module. No support is provided for multiple concurrent operators.

| Name | Type | Operator Type | Authentication Methods |
|---|---|---|---|
| Crypto Officer | Role | CO | None |

Table 13: Roles

## 4.3 Approved Services

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Message Digest | Compute a message digest | crypto_shash_init returns 0 | Message | Digest value | Message Digest | Crypto Officer |
| Encryption | Encrypt a plaintext | crypto_skcipher_setkey returns 0 | AES key, IV, plaintext | Ciphertext | Encryption, Decryption | Crypto Officer - AES Key: W,E |
| Decryption | Decrypt a ciphertext | crypto_skcipher_setkey returns 0 | AES key, IV, ciphertext | Plaintext | Encryption, Decryption | Crypto Officer - AES Key: W,E |
| Authenticated Encryption | Encrypt a plaintext | For all except AES GCM: crypto_aead_setkey returns 0; For AES GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set | AES key, IV, plaintext | Ciphertext, MAC tag | Authenticated Encryption, Authenticated Decryption | Crypto Officer - AES Key: W,E - HMAC Key: W,E |
| Authenticated Decryption | Decrypt a ciphertext | For all except AES GCM: crypto_aead_setkey returns 0; For AES GCM: crypto_aead_get_flags(tfm) has the CRYPTO_TFM_FIPS_COMPLIANCE flag set | AES key, IV, ciphertext, MAC tag | Plaintext | Authenticated Encryption, Authenticated Decryption | Crypto Officer - AES Key: W,E - HMAC Key: W,E |
| Message Authentication | Compute a MAC tag | crypto_shash_init returns 0 | AES key, message; HMAC key, message | MAC tag | Message Authentication | Crypto Officer - AES Key: W,E - HMAC Key: W,E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| Signature Verification | Verify a signature | crypto_akcipher_setkey returns 0 | RSA Public Key, Signature | Digital signature verification result | Digital Signature Verification | Crypto Officer<br>- RSA Public Key: W,E |
| Shared Secret Computation | Compute a shared secret | crypto_kpp_compute_shared_secret returns 0 | DH private key, DH public key; EC private key, EC public key | Shared secret | Shared Secret Computation | Crypto Officer<br>- DH Public Key: W,E<br>- DH Private Key: W,E<br>- Shared Secret: G,R |
| Key Pair Generation | Generate a key pair | crypto_kpp_set_secret and crypto_kpp_generate_public_key return 0 | Group; Curve | DH private key, DH public key; EC private key, EC public key | Key Pair Generation Safe Primes Key Generation Function | Crypto Officer<br>- DH Public Key: G,R<br>- DH Private Key: G,R<br>- Intermediate Key Generation Value: G,E,Z<br>- EC Public Key: G,R<br>- EC Private Key: G,R |
| Random Number Generation | Generate random bytes | crypto_rng_get_bytes returns 0 | Output length | Random bytes | Random Number Generation | Crypto Officer<br>- Entropy Input (IG D.L): W,E<br>- CTR_DRBG Seed (IG D.L): G,E<br>- Hash_DRBG Seed (IG D.L): G,E |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | - HMAC_DRBG Seed (IG D.L): G,E<br>- CTR_DRBG Internal State (V, Key) (IG D.L): G,E<br>- Hash_DRBG Internal State (V, C) (IG D.L): G,E<br>- HMAC_DRBG Internal State (V, Key) (IG D.L): G,E |
| Error Detection Code | Compute an EDC (crc32, crct10dif) | None | Message | EDC | None | Crypto Officer |
| Compression | Compress data (deflate, lz4, lz4hc, lzo, zlib-deflate, zstd) | None | Data | Compressed data | None | Crypto Officer |
| Generic System Call | Use the kernel to perform various non-cryptographic operations | None | Identifier, various arguments | Various return values | None | Crypto Officer |
| Show Version | Return the module name and version | None | N/A | Module name and version | None | Crypto Officer |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | information | | | | | |
| Show Status | Return the module status | None | N/A | Module status | None | Crypto Officer |
| Self-Test | Perform the CASTs and integrity tests | None | N/A | Pass/fail | Encryption, Decryption Authenticated Encryption, Authenticated Decryption Message Authentication Random Number Generation Key Pair Generation Shared Secret Computation Digital Signature Verification Message Digest Safe Primes Key Generation Function | Crypto Officer |
| Zeroization | Zeroize all SSPs | None | Any SSP | N/A | None | Crypto Officer<br>- AES Key: Z<br>- HMAC Key: Z<br>- Shared Secret: Z<br>- Entropy Input (IG D.L): Z<br>- |

| Name | Description | Indicator | Inputs | Outputs | Security Functions | SSP Access |
|---|---|---|---|---|---|---|
| | | | | | | CTR_DRBG Seed (IG D.L): Z<br>- Hash_DRBG Seed (IG D.L): Z<br>- HMAC_DRBG Seed (IG D.L): Z<br>- DH Public Key: Z<br>- DH Private Key: Z<br>- EC Public Key: Z<br>- EC Private Key: Z<br>- Intermediate Key Generation Value: Z<br>- CTR_DRBG Internal State (V, Key) (IG D.L): Z<br>- Hash_DRBG Internal State (V, C) (IG D.L): Z<br>- HMAC_DRBG Internal State (V, Key) (IG D.L): Z<br>- RSA Public Key: |

Table 14: Approved Services

The following convention is used to specify access rights to SSPs:

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

## 4.4 Non-Approved Services

| Name | Description | Algorithms | Role |
|---|---|---|---|
| AES GCM with external IV | Encryption | AES GCM with external IV | CO |
| KBKDF (libkcapi) | Key Derivation | KBKDF (libkcapi) | CO |
| HKDF (libkcapi) | Key Derivation | HKDF (libkcapi) | CO |
| PBKDF2 (libkcapi) | Password-Based Key Derivation | PBKDF2 (libkcapi) | CO |
| RSA | Encryption Primitive; Decryption Primitive | RSA | CO |
| RSA with PKCS#1 v1.5 padding | Signature Generation (pre-hashed message); Signature Verification (pre-hashed message); Key Encapsulation; Key Un-encapsulation | RSA with PKCS#1 v1.5 padding | CO |

Table 15: Non-Approved Services

## 4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

# 5 Software/Firmware Security

## 5.1 Integrity Techniques

The Linux kernel binary is integrity tested using an HMAC-SHA-512 calculation performed by the sha512hmac utility (which utilizes the module's HMAC and SHA-512 implementations) which compares the computed HMAC value with a precomputed HMAC value. An HMAC-SHA-512 calculation is also performed on the sha512hmac utility and the libkcapi library to verify their integrity by comparing the computed HMAC value with a precomputed HMAC value. The kernel crypto object files are loaded on start-up by the module and verified using RSA signature verification with PKCS#1 v1.5 padding, SHA-512, and a 4096-bit key.

## 5.2 Initiate on Demand

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

# 6 Operational Environment

## 6.1 Operational Environment Type and Requirements

**Type of Operational Environment**: Modifiable

**How Requirements are Satisfied**: 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.

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

# 7 Physical Security

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

# 8 Non-Invasive Security

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

# 9 Sensitive Security Parameters Management

## 9.1 Storage Areas

| Storage Area Name | Description | Persistence Type |
|---|---|---|
| RAM | Temporary storage for SSPs used by the module as part of service execution. The module does not perform persistent storage of SSPs | Dynamic |

Table 16: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form. 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 17: SSP Input-Output Methods

## 9.3 SSP Zeroization Methods

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

Table 18: SSP Zeroization Methods

All data output is inhibited during zeroization.

## 9.4 SSPs

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| AES Key | AES Key | XTS: 128, 256 bits; Other modes: 128, 192, 256 bits - XTS: 128, 256 bits; Other modes: 128, 192, 256 bits | Symmetric key - CSP | | | Encryption, Decryption Authenticated Encryption, Authenticated Decryption Message Authentication |
| HMAC Key | HMAC Key | 112-256 bits - 112-256 bits | Symmetric key - CSP | | | Authenticated Encryption, Authenticated Decryption Message Authentication |
| Shared Secret | Shared Secret | EC Diffie-Hellman P-256 and P-384 bits; Diffie-Hellman P-224-200 bits - EC Diffie-Hellman 128 and 192 bits; Diffie-Hellman 112-200 bits | Shared secret - CSP | | Shared Secret Computation | Shared Secret Computation |
| Entropy Input (IG D.L) | Entropy input used to seed the DRBGs | 128-384 bits - 128-384 bits | Entropy input - CSP | | | Random Number Generation |
| CTR_DRBG Seed (IG D.L) | DRBG seed derived from Entropy Input | 256, 320, 384 bits - 128, 192, 256 bits | Seed - CSP | Random Number Generation | | Random Number Generation |
| Hash_DRBG Seed (IG D.L) | DRBG seed derived from Entropy Input | 440, 888 bits - 128, 256 bits | Seed - CSP | Random Number Generation | | Random Number Generation |

| Name | Description | Size - Strength | Type - Category | Generated By | Established By | Used By |
|---|---|---|---|---|---|---|
| HMAC_DRBG Seed (IG D.L) | DRBG seed derived from Entropy Input | 440, 888 bits - 128, 256 bits | Seed - CSP | Random Number Generation | | Random Number Generation |
| CTR_DRBG Internal State (V, Key) (IG D.L) | Internal state of Counter DRBG instance | 256, 320, 384 bits - 128, 192, 256 bits | Internal state - CSP | Random Number Generation | | Random Number Generation |
| Hash_DRBG Internal State (V, C) (IG D.L) | Internal state of Hash DRBG instance | 880, 1776 bits - 128, 256 bits | Internal state - CSP | Random Number Generation | | Random Number Generation |
| HMAC_DRBG Internal State (V, Key) (IG D.L) | Internal state of HMAC DRBG instance | 320, 512, 1024 bits - 128, 256 bits | Internal state - CSP | Random Number Generation | | Random Number Generation |
| DH Public Key | DH Public Key | ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits | Public key - PSP | Safe Primes Key Generation Function | | Shared Secret Computation |
| DH Private Key | DH Private Key | - 112-200 bits | Private key - CSP | Safe Primes Key Generation Function | | Shared Secret Computation |
| EC Public Key | EC Public Key | P-256, P-384 bits - 128, 192 bits | Public key - PSP | Key Pair Generation | | Shared Secret Computation |
| EC Private Key | EC Private Key | P-256, P-384 bits - 128, 192 bits | Private key - CSP | Key Pair Generation | | Shared Secret Computation |
| Intermediate Key Generation Value | Intermediate Key Generation Value | 2048-8192 bits - 112-200 bits | Intermediate value - CSP | Key Pair Generation | | Key Pair Generation |
| RSA Public Key | RSA Public Key | 1024-16384 bits - 80-256 bits | Public key - PSP | | | Digital Signature Verification |

Table 19: SSP Table 1

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| AES Key | API input parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated Module Reset | |
| HMAC Key | API input parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated Module Reset | |
| Shared Secret | API output parameters | RAM:Plaintext | From service invocation to service completion | Automatic Module Reset | DH Public Key:Generated from DH Private Key:Generated from EC Public Key:Generated from EC Private Key:Generated from |
| Entropy Input (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Automatic Module Reset | CTR_DRBG Seed (IG D.L):Derives Hash_DRBG Seed (IG D.L):Derives HMAC_DRBG Seed (IG D.L):Derives |
| CTR_DRBG Seed (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Automatic Module Reset | Entropy Input (IG D.L):Derived From CTR_DRBG Internal State (V, Key) (IG D.L):Derives |
| Hash_DRBG Seed (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Automatic Module Reset | Entropy Input (IG D.L):Derived From Hash_DRBG Internal State (V, C) (IG D.L):Derives |
| HMAC_DRBG Seed (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Automatic Module Reset | Entropy Input (IG D.L):Derived From HMAC_DRBG Internal State (V, Key) (IG D.L):Derives |
| CTR_DRBG Internal State (V, Key) (IG D.L) | | RAM:Plaintext | From service invocation to | Wipe and Free memory block allocated | CTR_DRBG Seed (IG D.L):Derived From |

| Name | Input - Output | Storage | Storage Duration | Zeroization | Related SSPs |
|---|---|---|---|---|---|
| | | | service completion | | |
| Hash_DRBG Internal State (V, C) (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | Hash_DRBG Seed (IG D.L):Derived From |
| HMAC_DRBG Internal State (V, Key) (IG D.L) | | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | HMAC_DRBG Seed (IG D.L):Derived From |
| DH Public Key | API input parameters API output parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | Shared Secret:Paired With DH Private Key:Paired With |
| DH Private Key | API input parameters API output parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | Shared Secret:Paired With DH Public Key:Paired With |
| EC Public Key | API input parameters API output parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | Shared Secret:Paired With EC Private Key:Paired With |
| EC Private Key | API input parameters API output parameters | RAM:Plaintext | From service invocation to service completion | Wipe and Free memory block allocated | Shared Secret:Paired With EC Public Key:Paired With |
| Intermediate Key Generation Value | | RAM:Plaintext | From service invocation to service completion | Automatic | DH Public Key:Paired With DH Private Key:Paired With EC Public Key:Paired With EC Private Key:Paired With |
| RSA Public Key | API input parameters | | From service invocation to service completion | Automatic | |

Table 20: SSP Table 2

## 9.5 Transitions

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes, starting December 31, 2030.

The RSA algorithm as implemented by the module conforms to FIPS 186-4, which has been superseded by FIPS 186-5. FIPS 186-4 will be withdrawn on February 3, 2024.

# 10 Self-Tests

## 10.1 Pre-Operational Self-Tests

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details |
|---|---|---|---|---|---|
| HMAC-SHA2-512 (A4571) - x86 kernel | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for kernel binary |
| HMAC-SHA2-512 (A4571) - x86 libkcapi | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for libkcapi shared library |
| HMAC-SHA2-512 (A4571) - x86 sha512hmac | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for sha512hmac binary |
| RSA SigVer (FIPS186-4) (A4571) -x86 *.ko files | PKCS#1 v1.5 with SHA-512 4096-bit key | Signature Verification | SW/FW Integrity | Module becomes operational | Used for kernel object files |
| HMAC-SHA2-512 (A4557) - arm64 kernel | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for kernel binary |
| HMAC-SHA2-512 (A4557) - arm64 libkcapi | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for libkcapi shared library |
| HMAC-SHA2-512 (A4557) - arm64 sha512hmac | 128-bit key | Message Authentication | SW/FW Integrity | Module becomes operational | Used for sha512hmac binary |
| RSA SigVer (FIPS186-4) (A4551) - arm64 *.ko files | PKCS#1 v1.5 with SHA-512 4096-bit key | Signature Verification | SW/FW Integrity | Module becomes operational | Used for kernel object files |

Table 21: Pre-Operational Self-Tests

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

## 10.2 Conditional Self-Tests

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-CBC (A4563) - Encrypt | 128 bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4563) - Decrypt | 128 bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| AES-CBC (A4554) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4554) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4551) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4551) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4566) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4566) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4557) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4557) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4558) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4558) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC (A4561) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-CBC (A4561) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4554) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4554) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4551) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4551) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4566) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4566) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4557) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4557) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4558) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CBC-CS3 (A4558) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4554) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| AES-CCM (A4554) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4551) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4551) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4566) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4566) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4557) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4557) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4558) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CCM (A4558) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4554) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4554) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-CFB128 (A4551) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4551) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4566) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4566) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4558) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CFB128 (A4558) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4563) - Encrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4563) - Decrypt | 128-bit key | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4554) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4554) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4551) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4551) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| AES-CTR (A4566) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4566) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4557) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4557) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4558) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4558) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4561) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CTR (A4561) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4563) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4563) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4567) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-ECB (A4567) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4559) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4559) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4554) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4554) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4551) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4551) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4564) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4564) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4566) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4566) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4557) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| AES-ECB (A4557) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4568) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4568) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4558) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4558) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4556) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4556) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4560) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4560) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4561) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4561) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-ECB (A4552) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4552) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4555) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4555) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4553) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4553) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4565) - Encrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-ECB (A4565) - Decrypt | 128, 192, 256-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4563) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4563) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4567) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-GCM (A4567) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4559) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4559) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4554) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4554) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4551) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4551) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4564) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4564) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-GCM (A4566) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4566) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4568) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4568) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4558) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4558) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4556) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4556) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4560) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-GCM (A4560) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4552) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4552) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4555) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4555) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4553) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4553) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4565) - Encrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-GCM (A4565) - Decrypt | 128, 192, 256-bit keys, external IV and 96-bit internal IV, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| AES-OFB (A4554) - Encrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4554) - Decrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4551) - Encrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4551) - Decrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4566) - Encrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4566) - Decrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4558) - Encrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-OFB (A4558) - Decrypt | 128-bit keys, encrypt and decrypt | KAT | CAST | Module becomes operational | Symmetric operation | Test runs at power-on before the integrity test |
| AES-CMAC (A4554) | 128, 256-bit keys, generation | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| AES-CMAC (A4551) | 128, 256-bit keys, generation | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| AES-CMAC (A4566) | 128, 256-bit keys, generation | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| AES-CMAC (A4557) | 128, 256-bit keys, generation | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| AES-CMAC (A4558) | 128, 256-bit keys, generation | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| Counter DRBG (A4563) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4567) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4559) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4554) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4551) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4564) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4566) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | with prediction resistance | | | | | |
| Counter DRBG (A4568) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4558) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4556) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4560) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4552) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4555) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4553) | AES-128, AES-192, AES-256 without prediction resistance; AES-128 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Counter DRBG (A4565) | AES-128, AES-192, AES-256 without prediction | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | resistance; AES-128 with prediction resistance | | | | generate) health test | before the integrity test |
| Hash DRBG (A4569) | SHA-256, SHA-512 without prediction resistance; SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Hash DRBG (A4551) | SHA-256, SHA-512 without prediction resistance; SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Hash DRBG (A4571) | SHA-256, SHA-512 without prediction resistance; SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| Hash DRBG (A4570) | SHA-256, SHA-512 without prediction resistance; SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| HMAC DRBG (A4569) | HMAC-SHA-256, HMAC-SHA-512 without prediction resistance; HMAC-SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| HMAC DRBG (A4551) | HMAC-SHA-256, HMAC-SHA-512 without prediction resistance; HMAC-SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| HMAC DRBG (A4571) | HMAC-SHA-256, HMAC-SHA-512 without prediction resistance; HMAC-SHA-256 with | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | prediction resistance | | | | | |
| HMAC DRBG (A4570) | HMAC-SHA-256, HMAC-SHA-512 without prediction resistance; HMAC-SHA-256 with prediction resistance | KAT | CAST | Module becomes operational | Compliant with SP 800-90Ar1, (instantiate, reseed, generate) health test | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A4569) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A4551) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A4571) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A4557) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA-1 (A4570) | SHA-1 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4569) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4551) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4571) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4557) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| HMAC-SHA2-224 (A4562) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4570) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-224 (A4561) | SHA-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4569) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4551) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4571) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4557) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4562) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4570) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-256 (A4561) | SHA-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A4569) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A4551) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| HMAC-SHA2-384 (A4571) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A4557) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A4562) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-384 (A4570) | SHA-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4569) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4551) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4571) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4557) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4562) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA2-512 (A4570) | SHA-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-224 (A4551) | SHA3-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| HMAC-SHA3-224 (A4557) | SHA3-224 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-256 (A4551) | SHA3-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-256 (A4557) | SHA3-256 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-384 (A4551) | SHA3-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-384 (A4557) | SHA3-384 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-512 (A4551) | SHA3-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| HMAC-SHA3-512 (A4557) | SHA3-512 | KAT | CAST | Module becomes operational | Message authentication | Test runs at power-on before the integrity test |
| KAS-ECC-SSC Sp800-56Ar3 (A4551) | P-256, P-384 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |
| KAS-FFC-SSC Sp800-56Ar3 (A4551) | ffdhe2048 | KAT | CAST | Module becomes operational | Shared secret computation | Test runs at power-on before the integrity test |
| SHA-1 (A4569) | SHA-1 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA-1 (A4551) | SHA-1 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA-1 (A4571) | SHA-1 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| SHA-1 (A4557) | SHA-1 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA-1 (A4570) | SHA-1 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4569) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4551) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4571) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4557) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4562) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4570) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-224 (A4561) | SHA-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4569) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4551) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| SHA2-256 (A4571) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4557) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4562) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4570) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-256 (A4561) | SHA-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4569) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4551) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4571) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4557) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4562) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-384 (A4570) | SHA-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-512 (A4569) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| | | | | | | before the integrity test |
| SHA2-512 (A4551) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-512 (A4571) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-512 (A4557) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-512 (A4562) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA2-512 (A4570) | SHA-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-224 (A4551) | SHA3-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-224 (A4557) | SHA3-224 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-256 (A4551) | SHA3-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-256 (A4557) | SHA3-256 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-384 (A4551) | SHA3-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-384 (A4557) | SHA3-384 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |

| Algorithm or Test | Test Properties | Test Method | Test Type | Indicator | Details | Conditions |
|---|---|---|---|---|---|---|
| SHA3-512 (A4551) | SHA3-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| SHA3-512 (A4557) | SHA3-512 | KAT | CAST | Module becomes operational | Message digest | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-4) (A4569) | PKCS#1 v1.5 with 4096 bit key and SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-4) (A4551) | PKCS#1 v1.5 with 4096 bit key and SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-4) (A4571) | PKCS#1 v1.5 with 4096 bit key and SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| RSA SigVer (FIPS186-4) (A4570) | PKCS#1 v1.5 with 4096 bit key and SHA2-256 | KAT | CAST | Module becomes operational | Digital signature verification | Test runs at power-on before the integrity test |
| Safe Primes Key Generation (A4551) | N/A | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |
| ECDSA KeyGen (FIPS186-4) (A4551) | SHA2-256, P-256, P-384 curves, Appendix B.4.2 Testing Candidates | PCT | PCT | Successful key pair generation | Signature generation & verification | Key pair generation |

Table 22: Conditional Self-Tests

The module performs self-tests on all approved cryptographic algorithms as part of the approved services supported in the approved mode of operation, using the tests shown in the Conditional Self-Test table.

Upon generation of a DH or EC key pair, the module will perform a pair-wise consistency test (PCT), which provides some assurance that the generated key pair is well formed. This test consists of the PCT described in Section 5.6.2.1.4 of SP 800-56Ar3.

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

## 10.3 Periodic Self-Test Information

The module does not implement periodic self-tests.

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA2-512 (A4571) - x86 kernel | Message Authentication | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-512 (A4571) - x86 libkcapi | Message Authentication | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-512 (A4571) - x86 sha512hmac | Message Authentication | SW/FW Integrity | On demand | Manual |
| RSA SigVer (FIPS186-4) (A4571) -x86 *.ko files | Signature Verification | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-512 (A4557) - arm64 kernel | Message Authentication | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-512 (A4557) - arm64 libkcapi | Message Authentication | SW/FW Integrity | On demand | Manual |
| HMAC-SHA2-512 (A4557) - arm64 sha512hmac | Message Authentication | SW/FW Integrity | On demand | Manual |
| RSA SigVer (FIPS186-4) (A4551) - arm64 *.ko files | Signature Verification | SW/FW Integrity | On demand | Manual |

Table 23: Pre-Operational Periodic Information

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-CBC (A4563) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4563) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4557) - Encrypt | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-CBC (A4557) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4561) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC (A4561) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4557) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4557) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CBC-CS3 (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4557) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4557) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CCM (A4558) - Encrypt | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-CCM (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CFB128 (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4563) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4563) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4557) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4557) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4561) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-CTR (A4561) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4563) - Encrypt | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-ECB (A4563) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4567) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4567) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4559) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4559) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4564) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4564) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4557) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4557) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4568) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4568) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4556) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4556) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4560) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4560) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4561) - Encrypt | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-ECB (A4561) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4552) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4552) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4555) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4555) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4553) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4553) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4565) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-ECB (A4565) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4563) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4563) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4567) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4567) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4559) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4559) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4564) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4564) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4568) - Encrypt | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-GCM (A4568) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4556) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4556) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4560) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4560) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4552) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4552) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4555) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4555) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4553) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4553) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4565) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-GCM (A4565) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4554) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4554) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4551) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4551) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4566) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4566) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4558) - Encrypt | KAT | CAST | On Demand | Manually |
| AES-OFB (A4558) - Decrypt | KAT | CAST | On Demand | Manually |
| AES-CMAC (A4554) | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| AES-CMAC (A4551) | KAT | CAST | On Demand | Manually |
| AES-CMAC (A4566) | KAT | CAST | On Demand | Manually |
| AES-CMAC (A4557) | KAT | CAST | On Demand | Manually |
| AES-CMAC (A4558) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4563) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4567) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4559) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4554) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4551) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4564) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4566) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4568) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4558) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4556) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4560) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4552) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4555) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4553) | KAT | CAST | On Demand | Manually |
| Counter DRBG (A4565) | KAT | CAST | On Demand | Manually |
| Hash DRBG (A4569) | KAT | CAST | On Demand | Manually |
| Hash DRBG (A4551) | KAT | CAST | On Demand | Manually |
| Hash DRBG (A4571) | KAT | CAST | On Demand | Manually |
| Hash DRBG (A4570) | KAT | CAST | On Demand | Manually |
| HMAC DRBG (A4569) | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC DRBG (A4551) | KAT | CAST | On Demand | Manually |
| HMAC DRBG (A4571) | KAT | CAST | On Demand | Manually |
| HMAC DRBG (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A4569) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A4571) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA-1 (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4569) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4571) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4562) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-224 (A4561) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4569) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4571) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4562) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-256 (A4561) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A4569) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A4551) | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| HMAC-SHA2-384 (A4571) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A4562) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-384 (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4569) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4571) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4562) | KAT | CAST | On Demand | Manually |
| HMAC-SHA2-512 (A4570) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-224 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-224 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-256 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-256 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-384 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-384 (A4557) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-512 (A4551) | KAT | CAST | On Demand | Manually |
| HMAC-SHA3-512 (A4557) | KAT | CAST | On Demand | Manually |
| KAS-ECC-SSC Sp800-56Ar3 (A4551) | KAT | CAST | On Demand | Manually |
| KAS-FFC-SSC Sp800-56Ar3 (A4551) | KAT | CAST | On Demand | Manually |
| SHA-1 (A4569) | KAT | CAST | On Demand | Manually |
| SHA-1 (A4551) | KAT | CAST | On Demand | Manually |
| SHA-1 (A4571) | KAT | CAST | On Demand | Manually |
| SHA-1 (A4557) | KAT | CAST | On Demand | Manually |
| SHA-1 (A4570) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4569) | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| SHA2-224 (A4551) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4571) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4557) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4562) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4570) | KAT | CAST | On Demand | Manually |
| SHA2-224 (A4561) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4569) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4551) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4571) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4557) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4562) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4570) | KAT | CAST | On Demand | Manually |
| SHA2-256 (A4561) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4569) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4551) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4571) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4557) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4562) | KAT | CAST | On Demand | Manually |
| SHA2-384 (A4570) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4569) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4551) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4571) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4557) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4562) | KAT | CAST | On Demand | Manually |
| SHA2-512 (A4570) | KAT | CAST | On Demand | Manually |
| SHA3-224 (A4551) | KAT | CAST | On Demand | Manually |
| SHA3-224 (A4557) | KAT | CAST | On Demand | Manually |
| SHA3-256 (A4551) | KAT | CAST | On Demand | Manually |
| SHA3-256 (A4557) | KAT | CAST | On Demand | Manually |
| SHA3-384 (A4551) | KAT | CAST | On Demand | Manually |
| SHA3-384 (A4557) | KAT | CAST | On Demand | Manually |
| SHA3-512 (A4551) | KAT | CAST | On Demand | Manually |
| SHA3-512 (A4557) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-4) (A4569) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-4) (A4551) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-4) (A4571) | KAT | CAST | On Demand | Manually |
| RSA SigVer (FIPS186-4) (A4570) | KAT | CAST | On Demand | Manually |

| Algorithm or Test | Test Method | Test Type | Period | Periodic Method |
|---|---|---|---|---|
| Safe Primes Key Generation (A4551) | PCT | PCT | On Demand | Manually |
| ECDSA KeyGen (FIPS186-4) (A4551) | PCT | PCT | On Demand | Manually |

Table 24: Conditional Periodic Information

## 10.4 Error States

| Name | Description | Conditions | Recovery Method | Indicator |
|---|---|---|---|---|
| Error State | The Linux kernel immediately stops executing | Any self-test failure | Restart of the module | Kernel panic |

Table 25: Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the output interface is inhibited, and the module accepts no more inputs or requests (as the module is no longer running).

## 10.5 Operator Initiation of Self-Tests

The software integrity tests, CASTs and entropy source start-up tests can be invoked on demand by unloading and subsequently re-initializing the module. The PCTs can be invoked on demand by requesting the Key Pair Generation service.

# 11 Life-Cycle Assurance

## 11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the Amazon Linux 2023 package in the form of the kernel-6.1.41-64.118.amzn2023.rpm, kernel-6.1.41-64.118.fips.amzn2023.rpm, and libkcapi-hmaccalc-1.4.0-105.amzn2023.0.1.rpm packages.

## 11.2 Administrator Guidance

Before the RPM packages are installed, the Amazon Linux 2023 system must operate in the approved mode. This can be achieved by switching the system into the approved mode after the installation. Execute the fips-mode-setup --enable command. Restart the system. More information can be found in the vendor documentation.

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

After installation of the RPM packages, the Crypto Officer must execute the "cat /proc/sys/crypto/fips_name" command. The Crypto Officer must ensure that the proper name is listed in the output as follows:

Amazon Linux 2023 Kernel Cryptographic API

Then, the Crypto Officer must execute the "cat /proc/sys/crypto/fips_version" and "rpm -q libkcapi-hmaccalc" commands. These commands must output the following (one line per output):

**EC2 c7g.metal:**

6.1.41-64.118.amzn2023.aarch64

libkcapi-hmaccalc-1.4.0-105.amzn2023.0.1.aarch64

**EC2 c6i.metal:**

6.1.41-64.118.amzn2023.x86_64

libkcapi-hmaccalc-1.4.0-105.amzn2023.0.1.x86_64

**AWS Snowball, AWS Snowblade, AWS Snowcone:**

6.1.41-64.118.fips.amzn2023.x86_64

libkcapi-hmaccalc-1.4.0-105.amzn2023.0.1.x86_64

## 11.3 Non-Administrator Guidance

There is no non-administrator guidance.

## 11.6 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the RPM packages can be uninstalled from the Amazon Linux 2023 system.

# 12 Mitigation of Other Attacks

The module does not offer mitigation of other attacks and therefore this section is not applicable.

# Appendix A. Glossary and Abbreviations

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

| | |
|---|---|
| KBKDF | Key-based Key Derivation Function |
| KTS | Key Transport Scheme |
| KW | Key Wrap |
| MAC | Message Authentication Code |
| NIST | National Institute of Science and Technology |
| OFB | Output Feedback |
| PAA | Processor Algorithm Acceleration |
| PBKDF2 | Password-based Key Derivation Function v2 |
| PCT | Pair-Wise Consistency Test |
| PKCS | Public-Key Cryptography Standards |
| RSA | Rivest, Shamir, Adleman |
| SHA | Secure Hash Algorithm |
| SSC | Shared Secret Computation |
| SSP | Sensitive Security Parameter |
| XTS | XEX-based Tweaked-codebook mode with cipher text Stealing |

# Appendix B. References


| | |
|---|---|
| FIPS 140-3 | FIPS PUB 140-3 - Security Requirements For Cryptographic Modules<br>March 2019<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf |
| FIPS 140-3 IG | Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program: [09-02-2025]<br>https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements |
| FIPS 180-4 | Secure Hash Standard (SHS)<br>March 2012<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf |
| FIPS 186-4 | Digital Signature Standard (DSS)<br>July 2013<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf |
| FIPS 186-5 | Digital Signature Standard (DSS)<br>February 2023<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf |
| FIPS 197 | Advanced Encryption Standard<br>November 2001<br>https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf |
| FIPS 198-1 | The Keyed Hash Message Authentication Code (HMAC)<br>July 2008<br>https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf |
| FIPS 202 | SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions<br>August 2015<br>https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf |
| PKCS#1 | Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1<br>February 2003<br>https://www.ietf.org/rfc/rfc3447.txt |
| RFC 4106 | The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)<br>June 2005<br>https://datatracker.ietf.org/doc/html/rfc4106 |
| SP 800-38A | Recommendation for Block Cipher Modes of Operation Methods and Techniques<br>December 2001<br>https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf |
| SP 800-38A Addendum | Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode<br>October 2010<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf |
| SP 800-38B | Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication<br>May 2005<br>https://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf |

| | |
|---|---|
| SP 800-38C | Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality<br>May 2004<br>https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf |
| SP 800-38D | Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC<br>November 2007<br>https://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf |
| SP 800-38E | Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices<br>January 2010<br>https://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf |
| SP 800-38F | Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping<br>December 2012<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf |
| SP 800-56Ar3 | Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography<br>April 2018<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf |
| SP 800-90Ar1 | Recommendation for Random Number Generation Using Deterministic Random Bit Generators<br>June 2015<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf |
| SP 800-90B | Recommendation for the Entropy Sources Used for Random Bit Generation<br>January 2018<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf |
| SP 800-108r1 | NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions<br>August 2022<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1.pdf |
| SP 800-131Ar2 | Transitioning the Use of Cryptographic Algorithms and Key Lengths<br>March 2019<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf |
| SP 800-133r2 | Recommendation for Cryptographic Key Generation<br>June 2020<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf |
| SP 800-140Br1 | CMVP Security Policy Requirements<br>November 2023<br>https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf |