ORACLE
Linux

FIPS 140-3 Non-Proprietary Security Policy

Oracle Corporation

Oracle Linux 9 Kernel Crypto API Cryptographic Module

Software Version: kernel: 5.14.0-362.24.1.0.4.el9_3
libkcapi: 1.3.1-3.0.1.el9

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Title: Oracle Linux 9 Kernel Crypto API Cryptographic Module Security Policy
Date: June 30th, 2025
Contributing Authors:
Oracle Linux Engineering
Security Evaluations – Global Product Security
atsec information security

Oracle Corporation
World Headquarters
2300 Oracle Way
Austin, TX 78741
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

ORACLE

Table of Contents

ORACLE

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for Oracle Linux 9 Kernel Crypto API Cryptographic Module versions:

- kernel: 5.14.0-362.24.1.0.4.el9_3
- libkcapi: 1.3.1-3.0.1.el9

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.

1.1.1 How This Security Policy was prepared

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.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Oracle Linux 9 Kernel Crypto API Cryptographic Module (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: MultiChipStand

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the kernel binary and the loadable 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.

Tested Operational Environment’s Physical Perimeter (TOEPP):

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

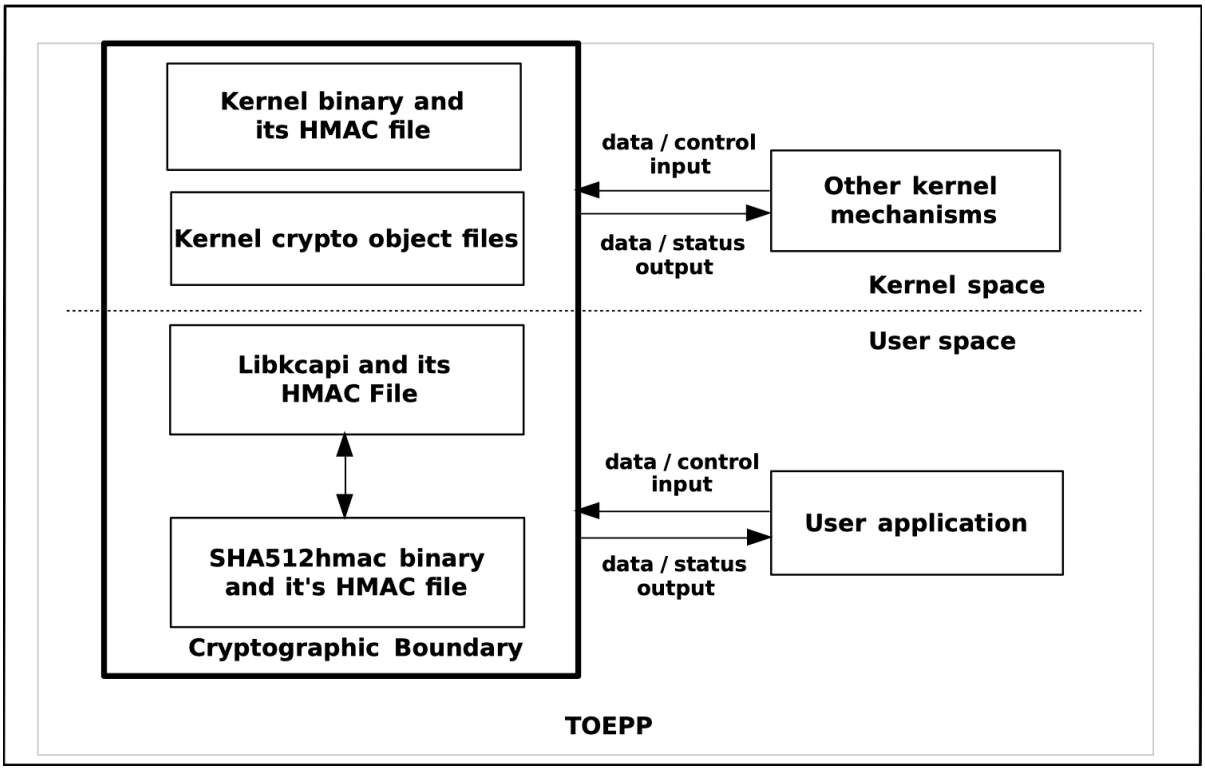

Figure 1: Block Diagram

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-5.14.0-362.24.1.0.4.el9_3.x86_64; *.ko and *.ko.xz files in /usr/lib/modules/5.14.0-362.24.1.0.4.el9_3.x86_64/kernel/crypto *.ko and *.ko.xz files in /usr/lib/modules/5.14.0-362.24.1.0.4.el9_3.x86_64/kernel/arch/x86/crypto; /usr/lib64/libkcapi.so.1.3.1, /usr/bin/sha512hmac	5.14.0-362.24.1.0.4.el9_3; 1.3.1-3.0.1.el9	N/A	HMAC-SHA-512; RSA signature verification

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)
Oracle Linux 9	ORACLE SERVER X9-2c	Intel Xeon Platinum 8358	Yes	KVM on Oracle Linux 8	5.14.0-362.24.1.0.4.el9_3; 1.3.1-3.0.1.el9
Oracle Linux 9	ORACLE SERVER E4-2c	AMD EPYC 7J13	Yes	KVM on Oracle Linux 8	5.14.0-362.24.1.0.4.el9_3; 1.3.1-3.0.1.el9

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

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

Operating System	Hardware Platform
Oracle Linux 9	Oracle X Series Servers
Oracle Linux 9	Oracle E Series Servers

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 within the cryptographic boundary excluded from the FIPS 140-3 requirements.

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.

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.

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5278, A5285, A5288	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A5282, A5293	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A5278, A5288	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A5280, A5291	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A5278, A5288	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A5278, A5285, A5288	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A5278, A5284, A5285, A5287, A5288, A5290	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A5283, A5286, A5289	Direction - Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A5278, A5288	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-OFB	A5281, A5292	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A5278, A5285, A5288	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
Hash DRBG	A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290, A5294, A5295, A5296	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290, A5294, A5295, A5296	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A5277, A5278, A5294, A5295, A5296, A5297	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5277, A5278, A5294, A5295, A5296, A5297	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5277, A5278, A5294, A5295, A5296, A5297	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5277, A5278, A5294, A5295, A5296	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5277, A5278, A5294, A5295, A5296	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA3-224	A5279	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A5279	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A5279	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A5279	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-FFC-SSC Sp800-56Ar3	A5277	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
RSA SigVer (FIPS186-4)	A5278, A5294, A5295, A5296	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A5278, A5294, A5295, A5296	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
Safe Primes Key Generation	A5277	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	SP 800-56A Rev. 3
SHA-1	A5277, A5278, A5294, A5295, A5296, A5297	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-224	A5277, A5278, A5294, A5295, A5296, A5297	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-256	A5277, A5278, A5294, A5295, A5296, A5297	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-384	A5277, A5278, A5294, A5295, A5296	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-512	A5277, A5278, A5294, A5295, A5296	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA3-224	A5279	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-256	A5279	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-384	A5279	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-512	A5279	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

N/A for this module.

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
KBKDF (libkcapi)	Key derivation
HKDF (libkcapi)	Key derivation
PBKDF2 (libkcapi)	Password-based key derivation
RSA	Encryption primitive; Decryption primitive
RSA with PKCS#1 v1.5 padding	Signature generation (pre-hashed message); Signature verification (pre-hashed message)

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
KAS-FFC-SSC	KAS-SSC	Shared Secret Computation	Keys:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 bit keys with 112-200 bits of key strength Compliance:Compliant with IG D.F scenario 2(1)	KAS-FFC-SSC Sp800-56Ar3: (A5277)
Safe Primes Key Generation	CKG	Key Generation	Groups:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	Safe Primes Key Generation: (A5277)
Encryption with AES	BC-UnAuth	Encrypt a plaintext with AES	Key size(s):128, 192, 256 bits (XTS mode 128 and 256 bits only)	AES-CBC: (A5278, A5285, A5288) AES-CBC-CS3: (A5282, A5293) AES-CFB128: (A5280, A5291) AES-CTR: (A5278, A5285, A5288) AES-ECB: (A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290) AES-OFB: (A5281, A5292) AES-XTS Testing Revision 2.0: (A5278, A5285, A5288)
Decryption with AES	BC-UnAuth	Decrypt a ciphertext with AES	Key size(s):128, 192, 256 bits (XTS mode 128 and 256 bits only)	AES-CBC: (A5278, A5285, A5288) AES-CBC-CS3: (A5282, A5293) AES-CFB128: (A5280, A5291) AES-CTR: (A5278, A5285, A5288) AES-ECB: (A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290) AES-OFB: (A5281, A5292) AES-XTS Testing Revision 2.0: (A5278, A5285, A5288)
Hashing	SHA	Compute a message digest	SHA-1:N/A SHA2-224:N/A SHA2-256:N/A SHA2-384:N/A SHA2-512:N/A SHA3-224:N/A SHA3-256:N/A SHA3-384:N/A SHA3-512:N/A	SHA-1: (A5277, A5278, A5294, A5295, A5296, A5297) SHA2-224: (A5277, A5278, A5294, A5295, A5296, A5297) SHA2-256: (A5277, A5278, A5294, A5295, A5296, A5297) SHA2-384: (A5277, A5278,

Name	Type	Description	Properties	Algorithms
				A5294, A5295, A5296) SHA2-512: (A5277, A5278, A5294, A5295, A5296) SHA3-224: (A5279) SHA3-256: (A5279) SHA3-384: (A5279) SHA3-512: (A5279)
Message authentication	MAC	Compute a MAC tag for authentication	HMAC key size(s):112-524288 bits (112-256 bits) AES key size(s):128, 192, 256 bits	AES-CMAC: (A5278, A5288) AES-GMAC: (A5278, A5288) HMAC-SHA-1: (A5277, A5278, A5294, A5295, A5296, A5297) HMAC-SHA2-224: (A5277, A5278, A5294, A5295, A5296, A5297) HMAC-SHA2-256: (A5277, A5278, A5294, A5295, A5296, A5297) HMAC-SHA2-384: (A5277, A5278, A5294, A5295, A5296, A5297) HMAC-SHA2-512: (A5277, A5278, A5294, A5295, A5296, A5297) HMAC-SHA3-224: (A5279) HMAC-SHA3-256: (A5279) HMAC-SHA3-384: (A5279) HMAC-SHA3-512: (A5279)
Random number generation with DRBGs	DRBG	Generate random numbers from DRBGs	Counter DRBG:128, 192, 256 bits HMAC DRBG:128, 256 bits Hash DRBG:128, 256 bits	Counter DRBG: (A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290) Hash DRBG: (A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290, A5294, A5295, A5296) HMAC DRBG: (A5277, A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290, A5294, A5295, A5296)
Signature verification with RSA	DigSig-SigVer	Verify a signature with RSA	Padding:PKCS#1 v1.5 Hashes:SHA1, SHA-224, SHA-256, SHA-384, SHA-512 Key size(s):2048, 3072, 4096 bits (112, 128, 150 bits)	RSA SigVer (FIPS186-4): (A5278, A5294, A5295, A5296) RSA SigVer (FIPS186-5): (A5278, A5294, A5295, A5296)
Authenticated encryption with AES	BC-Auth	Encrypt and authenticate a plaintext with AES	Key size(s):128, 192, 256 bits	AES-CCM: (A5278, A5288) AES-GCM: (A5278, A5283, A5284, A5285, A5286, A5287, A5288, A5289, A5290)
Authenticated decryption with AES	BC-Auth	Decrypt and authenticate a ciphertext with AES	Key size(s):128, 192, 256 bits	AES-CCM: (A5278, A5288) AES-GCM: (A5278, A5285, A5287, A5288, A5290, A5283, A5284, A5286, A5289)
Key wrapping	KTS-Wrap	Key wrapping	Key size(s):128, 192, 256 bits	AES-CCM: (A5278, A5288) AES-GCM: (A5283, A5285, A5286, A5289) AES-CBC: (A5278, A5285,

Name	Type	Description	Properties	Algorithms
				A5288) HMAC-SHA-1: (A5277, A5294, A5295, A5296, A5278) HMAC-SHA2-256: (A5294, A5295, A5296, A5278) HMAC-SHA2-384: (A5294, A5295, A5296, A5278) HMAC-SHA2-512: (A5294, A5295, A5296, A5278) AES-CTR: (A5278, A5285, A5288)
Key unwrapping	KTS-Wrap	Key unwrapping	Key sizes(s):128, 192, 256 bits	AES-CCM: (A5278, A5288) AES-GCM: (A5278, A5284, A5287, A5288, A5290) AES-CBC: (A5278, A5285, A5288) HMAC-SHA-1: (A5277, A5295, A5296, A5278, A5294) HMAC-SHA2-256: (A5295, A5296, A5278, A5294) HMAC-SHA2-384: (A5295, A5296, A5278, A5294) HMAC-SHA2-512: (A5295, A5296, A5278, A5294) AES-CTR: (A5278, A5285, A5288)

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

Legacy use:
Digital Signature Verification with SHA-1 is allowed for legacy use only.

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, as described in section 4.3.

2.7.2 AES XTS

The length of a single data unit encrypted or decrypted with AES XTS shall not exceed 2^{20} AES blocks, that is 16MB, of data per XTS instance. An XTS instance is defined in Section 4 of SP 800-38E. To meet the requirement stated in IG C.I, the module implements a check to ensure that the two AES keys used in AES XTS mode are not identical.

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

2.7.3 SP 800-56Arev3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the operator must use Diffie-Hellman shared secret computation algorithm with NVMe protocol. Additionally, the module's approved key pair generation service (see section 2.9) must be used to generate ephemeral 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, complying with Section 5.6.2.2.2 of SP 800-56Ar3. The module is compliant to IG D.F scenario 2 path (1).

2.7.4 RSA

For RSA signature verification, the module supports modulus sizes 2048, 3072, and 4096 bits compliant to IG C.F. All supported modulus sizes have been CAVP tested.

2.8 RBG and Entropy

Cert Number	Vendor Name
E151	Oracle Corporation

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Oracle Linux 9 Kernel CPU Time Jitter RNG Entropy Source	Non-Physical	Oracle Linux 9 on KVM on Oracle Linux 8 on Oracle SERVER X9-2c; Oracle Linux 9 on KVM on Oracle Linux 8 on ORACLE SERVER E4-2c	64 bits	57.46 bits	Linear-Feedback Shift Register (LFSR)

Table 10: 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-SHA2-512 DRBG implementation for internal purposes (e.g. to generate initialization vectors). This DRBG is initially seeded with 384 output bits from the entropy source (344 bits of entropy) and reseeded with 256 output bits from the entropy source (229 bits of entropy).

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133r2. When random values are required, they are obtained from the SP 800-90Ar1 approved DRBG, compliant with Section 4 of SP 800-133r2. The module implements safe primes key pair generation: compliant with SP 800-133r2, Section 5.2, which maps to SP 800-56Ar3. The method described in Section 5.6.1.1.4 of SP 800-56Ar3 ("Testing Candidates") is used. Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

2.10 Key Establishment

The module provides DH shared secret computation. Additionally, as permitted by IG D.G, the module provides key transport either by using an approved authenticated encryption mode or by a combination of any approved symmetric encryption mode and an approved authentication method.

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. No parts of this protocol, other than the AES GCM implementation, 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 11: 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

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 12: Roles

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

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message digest	Compute a message digest	crypto_shash_init returns 0	Message	Digest value	Hashing	Crypto Officer
Key wrapping	Wrap a key	crypto_skcipher_setkey returns 0; crypto_shash_init returns 0	AES key, key to be wrapped	wrapped key	Key wrapping	Crypto Officer - AES key: W,E - HMAC key: W,E
Key unwrapping	Unwrap a key	crypto_skcipher_setkey returns 0; crypto_shash_init returns 0	AES key, key to be unwrapped	unwrapped key	Key unwrapping	Crypto Officer - AES key: W,E - HMAC key: W,E
Encryption	Encrypt a plaintext	crypto_skcipher_setkey returns 0	AES key, plaintext	Ciphertext	Encryption with AES	Crypto Officer - AES key: W,E
Decryption	Decrypt a ciphertext	crypto_skcipher_setkey returns 0	AES key, ciphertext	Plaintext	Decryption with AES	Crypto Officer - AES key: W,E
Authenticated encryption	Encrypt and authenticate a plaintext	For all modes 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, plaintext	Ciphertext, MAC tag	Authenticated encryption with AES	Crypto Officer - AES key: W,E
Authenticated decryption	Encrypt and authenticate a ciphertext	For all modes 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, ciphertext, MAC tag	Plaintext or failure	Authenticated decryption with AES	Crypto Officer - AES key: W,E
Message authentication	Compute a MAC tag	crypto_shash_init returns 0	AES: AES key, message; HMAC: HMAC key, message	MAC tag	Message authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Shared Secret Computation	Compute a shared secret	crypto_kpp_compute_shared_secret returns 0	DH private key, DH remote public key	Shared secret	KAS-FFC-SSC	Crypto Officer - DH private key: W,E - DH remote public key: W,E - Shared secret: G,R
Key Pair Generation	Key Pair Generation	crypto_kpp_set_secret and crypto_kpp_generate_public_key return 0	Group	DH private key, DH remote public key	Safe Primes Key Generation	Crypto Officer - DH private key: G,R - DH remote public key: G,R - Intermediate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key Generation Value: G
Random number generation	Generate random bytes	crypto_rng_get_bytes returns 0	Output length	Random bytes	Random number generation with DRBGs	Crypto Officer - Entropy input: W,E - DRBG seed: G,E - DRBG Internal state (V, Key): G,E - DRBG Internal state (V, C): G,E
Error detection code	Compute an EDC (crc32, crct10dif)	None	Message	EDC	None	Crypto Officer
Compression	Compress data (deflate, lz4, lz4hc, lzo, zlibdeflate, 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 information	None	N/A	Module name and version	None	Crypto Officer
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 with AES Decryption with AES Hashing Message authentication Random number generation with DRBGs Signature verification with RSA Authenticated encryption with AES Authenticated decryption with AES KAS-FFC-SSC	Crypto Officer
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - Entropy input: Z - DRBG Internal state (V, Key): Z - DRBG Internal state

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						(V, C): Z - DRBG seed: Z - DH remote public key: Z - DH private key: Z

Table 13: Approved Services

The table above lists the 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.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM external IV encryption	Encrypt a plaintext using AES GCM with an external IV	AES GCM with external IV	CO
Key derivation	Derive a key from a key-derivation key or a shared secret	KBKDF (libkcapi) HKDF (libkcapi)	CO
Password-based key derivation	Derive a key from a password	PBKDF2 (libkcapi)	CO
RSA encryption primitive	Compute the raw RSA encryption of a plaintext	RSA	CO
RSA decryption primitive	Compute the raw RSA decryption of a cipertext	RSA	CO
RSA signature generation (pre-hashed message)	Generate a digital signature for a pre-hashed message	RSA with PKCS#1 v1.5 padding	CO
RSA signature verification (pre-hashed message)	Verify a digital signature for a pre-hashed message	RSA with PKCS#1 v1.5 padding	CO

Table 14: 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-SHA2-512 calculation performed by the sha512hmac utility (which utilizes the module's HMAC and SHA-512 implementations). An HMAC-SHA2-512 calculation is also performed on the sha512hmac utility and the libkcapi library to verify their integrity. The kernel crypto object files listed in section 2.2 are loaded on start-up by the module and verified using RSA signature verification with PKCS#1 v1.5 padding, SHA-256, and a 3072-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 11.1.

Instrumentation tools like the ptrace system call, gdb and strace, 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	Dynamic

Table 15: Storage Areas

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

9.2 SSP Input-Output Methods

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

Table 16: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the cipher handle	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the zeroization procedure succeeded.	By calling the appropriate zeroization functions: AES key: crypto_free_skcipher and crypto_free_aead; HMAC key: crypto_free_shash and crypto_free_ahash; DRBG internal state: crypto_free_rng; DRBG seed: crypto_free_rng; Entropy input string: crypto_free_rng; DH remote public & private key: crypto_free_kpp
Power off	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed. Module power off indicates that the zeroization procedure succeeded. The successful removal of power implicitly indicates that the zeroization is complete.	By removing power

Table 17: SSP Zeroization Methods

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key used for encryption, decryption, and computing MAC tags.	128, 192, 256 bits - 128, 192, 256 bits	Symmetric Key - CSP			Encryption with AES Decryption with AES Authenticated encryption with AES

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Authenticated decryption with AES Key wrapping Key unwrapping
HMAC key	HMAC key.	112-524288 bits - 112-256 bits	Authentication key - CSP			Message authentication Key wrapping Key unwrapping
Entropy input	Entropy input used to seed the DRBGs. Compliant with IG D.L.	128-384 bits - 117-357 bits	Entropy input - CSP			Random number generation with DRBGs
DRBG seed	DRBG seed derived from entropy input. Compliant with IG D.L.	CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRBG: 128, 256 bits - CTR_DRBG: 128, 192, 256 bits; Hash_DRBG: 128, 256 bits; HMAC_DRBG: 128, 256 bits	Seed - CSP	Random number generation with DRBGs		Random number generation with DRBGs
DRBG Internal state (V, Key)	Internal state of CTR_DRBG and HMAC_DRBG instances. Compliant with IG D.L.	CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256 bits	Internal state - CSP	Random number generation with DRBGs		Random number generation with DRBGs
DRBG Internal state (V, C)	Internal state of Hash_DRBG instances. Compliant with IG D.L.	Hash_DRBG: 128, 256 bits - Hash_DRBG: 128, 256 bits	Internal state - CSP	Random number generation with DRBGs		Random number generation with DRBGs
Intermediate Key Generation Value	Temporary value generated during Key Pair Generation services.	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, P-256, P-384 - 112-200 bits	Intermediate value - CSP	Safe Primes Key Generation		Safe Primes Key Generation
DH remote public key	Public key used for Diffie-Hellman.	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Public key - PSP	Safe Primes Key Generation		KAS-FFC-SSC Safe Primes Key Generation
DH private key	Private key used for Diffie-Hellman.	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Private key - CSP	Safe Primes Key Generation		KAS-FFC-SSC Safe Primes Key Generation
Shared secret	Shared secret generated by DH.	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Shared secret - CSP		KAS-FFC-SSC	

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	Until cipher handled is freed or module powered off	Free cipher handle Power off	
HMAC key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	Until cipher handled is freed or module powered off	Free cipher handle Power off	
Entropy input		RAM:Plaintext	From generation until DRBG seed/reseed	Free cipher handle Power off	DRBG seed:Derives
DRBG seed		RAM:Plaintext	While the DRBG is being instantiated	Free cipher handle Power off	Entropy input:Derived From DRBG Internal state (V, Key):Derives DRBG Internal state (V, C):Derives
DRBG Internal state (V, Key)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Power off	DRBG seed:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG Internal state (V, C)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Power off	DRBG seed:Derived From
Intermediate Key Generation Value		RAM:Plaintext	Until key pair generation service completes.	Free cipher handle Power off	DH private key:Derived From DH remote public key:Derived From
DH remote public key	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	Until cipher handled is freed or module powered off	Free cipher handle Power off	DH private key:Used With
DH private key	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	Until cipher handled is freed or module powered off	Free cipher handle Power off	DH remote public key:Used With
Shared secret	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	Until cipher handled is freed or module powered off	Free cipher handle Power off	DH private key:Used With DH remote public key:Used With

Table 19: SSP Table 2

9.5 Transitions

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

The RSA algorithm with SHA-1 as implemented by the module conforms to FIPS 186-4. FIPS 186-4 was withdrawn on February 3, 2024 but FIPS 140-3 IG C.K allows RSA signature verification with SHA-1 under FIPS 186-4 to still be approved.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-512 (A5296)	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use.	Integrity test for vmlinuz, libkcapi components and sha512hmac binary
RSA SigVer (FIPS186-4) (A5278)	3072-bit key with SHA-256	Signature Verification	SW/FW Integrity	Module becomes operational and services are available for use.	Integrity test for kernel object files

Table 20: 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. The algorithms used for the integrity test (i.e., HMAC-SHA2-512 and RSA SigVer with 3072 bit key) run their CASTs before the integrity test is performed. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the pre-operational software integrity self-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
Safe Primes Key Generation (A5277)	N/A	PCT	PCT	crypto_kpp_generate_public_key returns 0	SP 800-56Ar3 Section 5.6.2.1.4	Key pair generation
SHA-1 (A5278)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA-1 (A5294)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA-1 (A5295)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA-1 (A5296)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5278)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5294)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5295)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5296)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A5278)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A5294)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256 (A5295)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A5296)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A5278)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A5294)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A5295)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A5296)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A5278)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A5294)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A5295)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A5296)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-224 (A5279)	0, 8, 448 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-256 (A5279)	0, 8, 448 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-384 (A5279)	0, 8, 448 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA3-512 (A5279)	0, 8, 448 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
AES-ECB (A5278)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5283)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5284)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5285)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5286)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5287)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A5289)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5290)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5277)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-ECB (A5288)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CBC (A5278)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CBC (A5285)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CBC-CS3 (A5293)	128 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-OFB (A5292)	128 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CFB128 (A5291)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CTR (A5278)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CTR (A5288)	128, 192, 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CCM (A5288)	128, 192, 256 bit keys; 128-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5278)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5283)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM (A5284)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5285)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5286)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization
AES-GCM (A5287)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5288)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-GCM (A5289)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A5290)	128, 192, 256 bit keys, 96-bit IVs	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-XTS Testing Revision 2.0 (A5278)	128 and 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-XTS Testing Revision 2.0 (A5288)	128 and 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Encryption, Decryption	Module initialization
AES-CMAC (A5288)	128 and 256 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A5278)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A5294)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A5295)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A5296)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5278)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5294)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5295)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5296)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5278)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5294)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5295)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5296)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A5278)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A5294)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A5295)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A5296)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-512 (A5278)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA2-512 (A5294)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA3-224 (A5279)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-512 (A5295)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA2-512 (A5296)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization. Before integrity test.
HMAC-SHA3-256 (A5279)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA3-384 (A5279)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA3-512 (A5279)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
KAS-FFC-SSC Sp800-56Ar3 (A5277)	ffdhe2048	KAT	CAST	Module becomes operational and services are available for use.	Shared secret computation	Module initialization
Counter DRBG (A5278)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5283)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5284)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5285)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5286)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5287)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5288)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5289)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Counter DRBG (A5290)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5278)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5283)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Hash DRBG (A5284)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5285)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5286)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5287)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5288)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5289)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5290)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5294)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5295)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5296)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5278)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5283)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5284)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5285)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5286)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5287)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5288)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5289)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5290)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5294)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG (A5295)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5296)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
RSA SigVer (FIPS186-5) (A5278)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Verify	Module initialization. Before integrity test.
RSA SigVer (FIPS186-5) (A5294)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Verify	Module initialization. Before integrity test.
RSA SigVer (FIPS186-5) (A5295)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Verify	Module initialization. Before integrity test.
RSA SigVer (FIPS186-5) (A5296)	4096-bit key with SHA-256	KAT	CAST	Module becomes operational and services are available for use.	Verify	Module initialization. Before integrity test.
Entropy Source (RCT Start-Up)	1024 samples	RCT	CAST	Module becomes operational and services are available for use.	Entropy source start-up test	Entropy source initialization
Entropy Source (APT Start-Up)	1024 samples	APT	CAST	Module becomes operational and services are available for use.	Entropy source start-up test	Entropy source initialization
Entropy Source (RCT Runtime)	Cutoff C = 61	RCT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
Entropy Source (APT Runtime)	Cutoff C = 355	APT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
Counter DRBG (A5277)	128, 192, 256 bit keys With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
Hash DRBG (A5277)	SHA-256 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC DRBG (A5277)	SHA-256, SHA512 With/without PR; Health test per section 11.3 of SP 800-90Arev1	KAT	CAST	Module becomes operational and services are available for use.	Seed, Generate	Module initialization
HMAC-SHA-1 (A5277)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA-1 (A5297)	160, 200 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5277)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-224 (A5297)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5277)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-256 (A5297)	256, 296 and 640 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
HMAC-SHA2-384 (A5277)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization
SHA-1 (A5277)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A5297)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5277)	0, 24, 448, 512, 1304 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-224 (A5297)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A5277)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-256 (A5297)	0, 24, 448, 512 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-384 (A5277)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
SHA2-512 (A5277)	0, 24, 448, 832, 896 and 8184 bit messages	KAT	CAST	Module becomes operational and services are available for use.	Message digest	Module initialization
HMAC-SHA2-512 (A5277)	160 and 1048 bit keys	KAT	CAST	Module becomes operational and services are available for use.	Message authentication	Module initialization

Table 21: 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 table above. Services are not available, and data output (via the data output interface) is inhibited during the self-tests. If any of these tests fails, the module transitions to the Error state.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A5296)	Message Authentication	SW/FW Integrity	On demand	Manually
RSA SigVer (FIPS186-4) (A5278)	Signature Verification	SW/FW Integrity	On demand	Manually

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Safe Primes Key Generation (A5277)	PCT	PCT	On demand	Manually
SHA-1 (A5278)	KAT	CAST	On demand	Manually
SHA-1 (A5294)	KAT	CAST	On demand	Manually
SHA-1 (A5295)	KAT	CAST	On demand	Manually
SHA-1 (A5296)	KAT	CAST	On demand	Manually
SHA2-224 (A5278)	KAT	CAST	On demand	Manually
SHA2-224 (A5294)	KAT	CAST	On demand	Manually
SHA2-224 (A5295)	KAT	CAST	On demand	Manually
SHA2-224 (A5296)	KAT	CAST	On demand	Manually
SHA2-256 (A5278)	KAT	CAST	On demand	Manually
SHA2-256 (A5294)	KAT	CAST	On demand	Manually
SHA2-256 (A5295)	KAT	CAST	On demand	Manually
SHA2-256 (A5296)	KAT	CAST	On demand	Manually
SHA2-384 (A5278)	KAT	CAST	On demand	Manually
SHA2-384 (A5294)	KAT	CAST	On demand	Manually
SHA2-384 (A5295)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-384 (A5296)	KAT	CAST	On demand	Manually
SHA2-512 (A5278)	KAT	CAST	On demand	Manually
SHA2-512 (A5294)	KAT	CAST	On demand	Manually
SHA2-512 (A5295)	KAT	CAST	On demand	Manually
SHA2-512 (A5296)	KAT	CAST	On demand	Manually
SHA3-224 (A5279)	KAT	CAST	On demand	Manually
SHA3-256 (A5279)	KAT	CAST	On demand	Manually
SHA3-384 (A5279)	KAT	CAST	On demand	Manually
SHA3-512 (A5279)	KAT	CAST	On demand	Manually
AES-ECB (A5278)	KAT	CAST	On demand	Manually
AES-ECB (A5283)	KAT	CAST	On demand	Manually
AES-ECB (A5284)	KAT	CAST	On demand	Manually
AES-ECB (A5285)	KAT	CAST	On demand	Manually
AES-ECB (A5286)	KAT	CAST	On demand	Manually
AES-ECB (A5287)	KAT	CAST	On demand	Manually
AES-ECB (A5289)	KAT	CAST	On demand	Manually
AES-ECB (A5290)	KAT	CAST	On demand	Manually
AES-ECB (A5277)	KAT	CAST	On demand	Manually
AES-ECB (A5288)	KAT	CAST	On demand	Manually
AES-CBC (A5278)	KAT	CAST	On demand	Manually
AES-CBC (A5285)	KAT	CAST	On demand	Manually
AES-CBC-CS3 (A5293)	KAT	CAST	On demand	Manually
AES-OFB (A5292)	KAT	CAST	On demand	Manually
AES-CFB128 (A5291)	KAT	CAST	On demand	Manually
AES-CTR (A5278)	KAT	CAST	On demand	Manually
AES-CTR (A5288)	KAT	CAST	On demand	Manually
AES-CCM (A5288)	KAT	CAST	On demand	Manually
AES-GCM (A5278)	KAT	CAST	On demand	Manually
AES-GCM (A5283)	KAT	CAST	On demand	Manually
AES-GCM (A5284)	KAT	CAST	On demand	Manually
AES-GCM (A5285)	KAT	CAST	On demand	Manually
AES-GCM (A5286)	KAT	CAST	On demand	Manually
AES-GCM (A5287)	KAT	CAST	On demand	Manually
AES-GCM (A5288)	KAT	CAST	On demand	Manually
AES-GCM (A5289)	KAT	CAST	On demand	Manually
AES-GCM (A5290)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (A5278)	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (A5288)	KAT	CAST	On demand	Manually
AES-CMAC (A5288)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5278)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5294)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5295)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5296)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A5278)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A5294)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A5295)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A5296)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5278)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5294)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5295)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5296)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A5278)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A5294)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A5295)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A5296)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A5278)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A5294)	KAT	CAST	On demand	Manually
HMAC-SHA3-224 (A5279)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A5295)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A5296)	KAT	CAST	On demand	Manually
HMAC-SHA3-256 (A5279)	KAT	CAST	On demand	Manually
HMAC-SHA3-384 (A5279)	KAT	CAST	On demand	Manually
HMAC-SHA3-512 (A5279)	KAT	CAST	On demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A5277)	KAT	CAST	On demand	Manually
Counter DRBG (A5278)	KAT	CAST	On demand	Manually
Counter DRBG (A5283)	KAT	CAST	On demand	Manually
Counter DRBG (A5284)	KAT	CAST	On demand	Manually
Counter DRBG (A5285)	KAT	CAST	On demand	Manually
Counter DRBG (A5286)	KAT	CAST	On demand	Manually
Counter DRBG (A5287)	KAT	CAST	On demand	Manually
Counter DRBG (A5288)	KAT	CAST	On demand	Manually
Counter DRBG (A5289)	KAT	CAST	On demand	Manually
Counter DRBG (A5290)	KAT	CAST	On demand	Manually
Hash DRBG (A5278)	KAT	CAST	On demand	Manually
Hash DRBG (A5283)	KAT	CAST	On demand	Manually
Hash DRBG (A5284)	KAT	CAST	On demand	Manually
Hash DRBG (A5285)	KAT	CAST	On demand	Manually
Hash DRBG (A5286)	KAT	CAST	On demand	Manually
Hash DRBG (A5287)	KAT	CAST	On demand	Manually
Hash DRBG (A5288)	KAT	CAST	On demand	Manually
Hash DRBG (A5289)	KAT	CAST	On demand	Manually
Hash DRBG (A5290)	KAT	CAST	On demand	Manually
Hash DRBG (A5294)	KAT	CAST	On demand	Manually
Hash DRBG (A5295)	KAT	CAST	On demand	Manually
Hash DRBG (A5296)	KAT	CAST	On demand	Manually
HMAC DRBG (A5278)	KAT	CAST	On demand	Manually
HMAC DRBG (A5283)	KAT	CAST	On demand	Manually
HMAC DRBG (A5284)	KAT	CAST	On demand	Manually
HMAC DRBG (A5285)	KAT	CAST	On demand	Manually
HMAC DRBG (A5286)	KAT	CAST	On demand	Manually
HMAC DRBG (A5287)	KAT	CAST	On demand	Manually
HMAC DRBG (A5288)	KAT	CAST	On demand	Manually
HMAC DRBG (A5289)	KAT	CAST	On demand	Manually
HMAC DRBG (A5290)	KAT	CAST	On demand	Manually
HMAC DRBG (A5294)	KAT	CAST	On demand	Manually
HMAC DRBG (A5295)	KAT	CAST	On demand	Manually
HMAC DRBG (A5296)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A5278)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A5294)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A5295)	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5) (A5296)	KAT	CAST	On demand	Manually
Entropy Source (RCT Start-Up)	RCT	CAST	On demand	Manually
Entropy Source (APT Start-Up)	APT	CAST	On demand	Manually
Entropy Source (RCT Runtime)	RCT	CAST	On demand	Manually
Entropy Source (APT Runtime)	APT	CAST	On demand	Manually
Counter DRBG (A5277)	KAT	CAST	On demand	Manually
Hash DRBG (A5277)	KAT	CAST	On demand	Manually
HMAC DRBG (A5277)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5277)	KAT	CAST	On demand	Manually
HMAC-SHA-1 (A5297)	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (A5277)	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-224 (A5297)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5277)	KAT	CAST	On demand	Manually
HMAC-SHA2-256 (A5297)	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (A5277)	KAT	CAST	On demand	Manually
SHA-1 (A5277)	KAT	CAST	On demand	Manually
SHA-1 (A5297)	KAT	CAST	On demand	Manually
SHA2-224 (A5277)	KAT	CAST	On demand	Manually
SHA2-224 (A5297)	KAT	CAST	On demand	Manually
SHA2-256 (A5277)	KAT	CAST	On demand	Manually
SHA2-256 (A5297)	KAT	CAST	On demand	Manually
SHA2-384 (A5277)	KAT	CAST	On demand	Manually
SHA2-512 (A5277)	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (A5277)	KAT	CAST	On demand	Manually

Table 23: 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 24: Error States

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, cryptographic algorithm self-tests, and entropy source start-up tests can be invoked on demand by unloading and subsequently re-initializing the module. The pair-wise consistency tests can be invoked on demand by requesting the key pair generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as part of the Oracle Linux 9 (OL9) RPM packages in the form of kernel-5.14.0-362.24.1.0.4.el9_3, libkcapi-1.3.1-3.0.1.el9 and libkcapi-hmaccalc-1.3.1-3.0.1.el9 packages that is located in the "Oracle Linux 9 Security Validation (Update 3)" yum repository (ol9_u3_security_validation).

The operational environment needs to be set up in the FIPS validated configuration by installing the module as follows:

- For installation, add the fips=1 option to the kernel command line during the system installation. During the software selection stage, do not install any third-party software.
- Switching the system into the FIPS validated configuration after the installation, execute the fips-mode-setup --enable command. Restart the system using the reboot command.

In both cases, the Crypto Officer must verify the OL8 and OL9 systems operate in the approved mode by executing the fips-mode-setup --check command, which should output "FIPS mode is enabled."

More information can be found at the vendor's documentation.

11.2 Administrator Guidance

After installation of the kernel-5.14.0-362.24.1.0.4.el9_3, libkcapi-1.3.1-3.0.1.el9 and libkcapi-hmaccalc-1.3.1-3.0.1.el9 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:

Oracle Linux 9 Kernel Crypto API Cryptographic Module

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

5.14.0-362.24.1.0.4.el9_3.x86_64

libkcapi-1.3.1-3.0.1.el9.x86_64

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the kernel-5.14.0-362.24.1.0.4.el9_3, libkcapi-1.3.1-3.0.1.el9 and libkcapi-hmaccalc-1.3.1-3.0.1.el9 RPM packages can be uninstalled from the Oracle Linux 9 systems.

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
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECDH	Elliptic Curve Diffie-Hellman
ECDH	Elliptic Curve Diffie-Hellman
ENT (NP)	Non-physical Entropy Source
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
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
PAA	Processor Algorithm Acceleration
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

ANS X9.42-2001	Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography 2001 https://webstore.ansi.org/standards/ascx9/ansix9422001
ANS X9.63-2001	Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography 2001 https://webstore.ansi.org/standards/ascx9/ansix9632001
FIPS 140-3	FIPS PUB 140-3 - Security Requirements For Cryptographic Modules March 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG	Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	Secure Hash Standard (SHS) March 2012 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS 186-4	Digital Signature Standard (DSS) July 2013 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS 197	Advanced Encryption Standard November 2001 https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS 198-1	The Keyed Hash Message Authentication Code (HMAC) July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS 202	SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 February 2003 https://www.ietf.org/rfc/rfc3447.txt
RFC 3526	More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE) May 2003 https://www.ietf.org/rfc/rfc3526.txt
RFC 5288	AES Galois Counter Mode (GCM) Cipher Suites for TLS August 2008 https://www.ietf.org/rfc/rfc5288.txt
RFC 7919	Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS) August 2016 https://www.ietf.org/rfc/rfc7919.txt

RFC 8446	The Transport Layer Security (TLS) Protocol Version 1.3 August 2018 https://www.ietf.org/rfc/rfc8446.txt
SP 800-38A	Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001 https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP 800-38A Addendum	Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode October 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf
SP 800-38B	Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication May 2005 https://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 May 2004 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP 800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC November 2007 https://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP 800-38E	Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices January 2010 https://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP 800-38F	Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping December 2012 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP 800-52r2	Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations August 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf
SP 800-56Ar3	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography April 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
SP 800-56Cr2	Recommendation for Key-Derivation Methods in Key-Establishment Schemes August 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf
SP 800-90Ar1	Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

SP 800-90B	Recommendation for the Entropy Sources Used for Random Bit Generation January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP 800-108r1	NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions August 2022 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1.pdf
SP 800-131Ar2	Transitioning the Use of Cryptographic Algorithms and Key Lengths March 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf
SP 800-132	Recommendation for Password-Based Key Derivation - Part 1: Storage Applications December 2010 https://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf
SP 800-133r2	Recommendation for Cryptographic Key Generation June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP 800-135r1	Recommendation for Existing Application-Specific Key Derivation Functions December 2011 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf
SP 800-140Br1	CMVP Security Policy Requirements November 2023 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf