IBM Corporation

IBM COS Linux Kernel Cryptographic API Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.1

Last update: 2025-11-10

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

Prepared for:
IBM Corporation
71 S Wacker Dr, 6th Floor
Chicago, IL 60606
www.ibm.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 3 of the IBM COS Linux Kernel Cryptographic API Cryptographic Module. 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

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

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 IBM COS Linux Kernel Cryptographic 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
Module Characteristics:

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 kcapi-hasher 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 indicated by the small bold border 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. It includes software in kernel and user space, as well as the PAA in the CPU. The TOEPP is indicated by the large thin border in Figure 1.

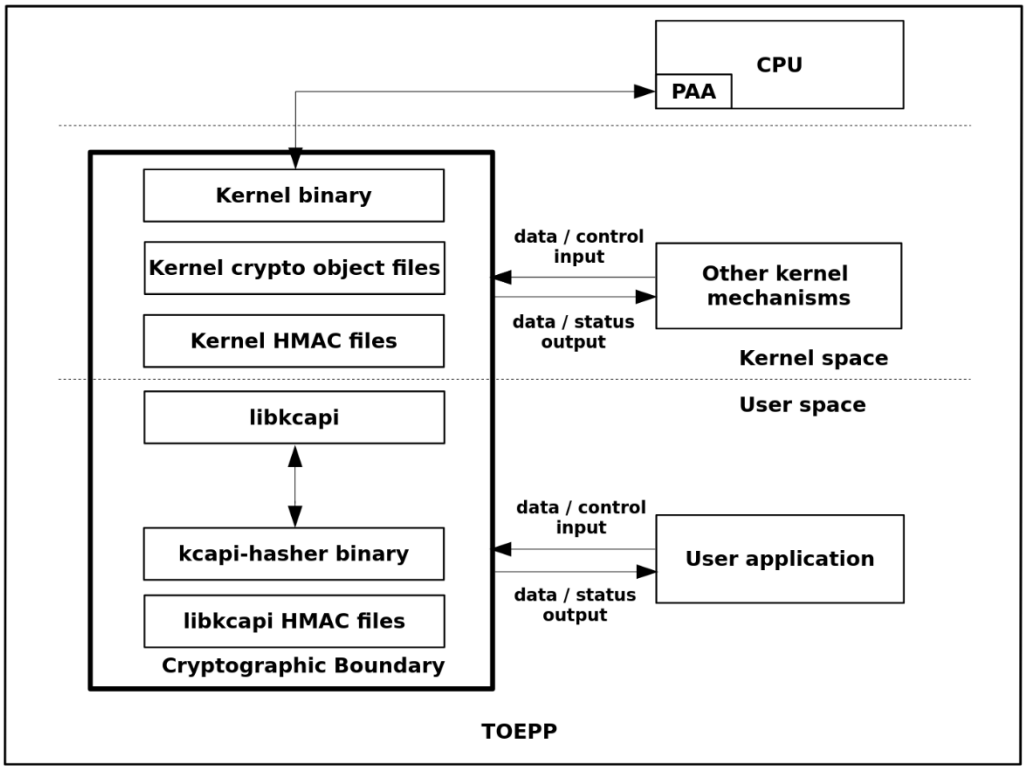

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:
N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
/boot/vmlinuz-6.1.0-32-amd64; *.ko files in /usr/lib/modules/6.1.0-32-amd64/kernel/crypto/; *.ko files in /usr/lib/modules/6.1.0-32-amd64/kernel/arch/x86/crypto/; /usr/lib/x86_64-linux-gnu/libkcapi.so.1.5.0; /usr/bin/kcapi-hasher	Kernel: 3; libkcapi: 1.5.0-1 amd64	N/A	HMAC SHA-512 (vmlinuz, libkcapi.so.1.5.0, kcapi-hasher); RSA signature verification (*.ko files)

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

Tested Module Identification – Hybrid Disjoint Hardware:
N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

The module makes use of hardware acceleration provided by the hardware platform. Namely, AES-NI from the platform listed in the *Tested Operational Environments - Software, Firmware, Hybrid* table. AES-NI and SHA extensions are considered as PAA.

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
ClevOS 3.19	Lenovo SR650v3	Intel Sapphire Rapids Xeon 8474C	Yes	N/A	Kernel: 3 libkcapi: 1.5.0-1 amd64
ClevOS 3.19	Lenovo SR650v3	Intel Sapphire Rapids Xeon 8474C	No	N/A	Kernel: 3 libkcapi: 1.5.0-1 amd64

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

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

Operating System	Hardware Platform
ClevOS 3.19	PIO-628U-TR4T+-ST031 (Intel Xeon E5-2620 *)
ClevOS 3.19	PIO-648R-E1CR36L+-ST031 (Intel Xeon E5-2620 *)
ClevOS 3.19	IBM A10 Series (Intel Xeon 6126)
ClevOS 3.19	IBM A10 Series (Intel Xeon 6226)
ClevOS 3.19	IBM M10 Series (Intel Xeon 4110)
ClevOS 3.19	IBM M10 Series (Intel Xeon 4210R)
ClevOS 3.19	IBM C10 Series (Intel Xeon 4110)

Operating System	Hardware Platform
ClevOS 3.19	IBM C10 Series (Intel Xeon 4210R)
ClevOS 3.19	IBM 4616-A2D Series (Intel Xeon 4416+)
ClevOS 3.19	IBM 4616-M2D Series (Intel Xeon 4416+)
ClevOS 3.19	IBM 4616-C2D Series (Intel Xeon 4416+)
ClevOS 3.19	IBM 4616-S3D Series (Intel Xeon Gold 6438N)
ClevOS 3.19	IBM 4616-S4D/S6D Series (Intel Xeon 4416+)
ClevOS 3.19	IBM 4616-A1D Series (Intel Xeon 4314)
ClevOS 3.19	IBM 4616-M1D Series (Intel Xeon 4314)
ClevOS 3.19	IBM 4616-C1D Series (Intel Xeon 4314)
ClevOS 3.19	IBM 4616-S2D Series (Intel Xeon 4314)

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

Not applicable.

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

Mode Change Instructions and Status:

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. 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:

Not applicable.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6801, A6806, A6809, A6812	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A6801, A6806, A6809, A6812	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A6801, A6806, A6812	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A6801, A6806, A6812	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A6801, A6806, A6812	Direction - Generation Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A6801, A6806, A6809, A6812	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A6801, A6804, A6805, A6806, A6807, A6808, A6809, A6810, A6811, A6812, A6813, A6814	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A6801, A6805, A6806, A6808, A6809, A6811, A6812, A6814	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A6804, A6807, A6810, A6813	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A6801, A6806, A6812	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-KW	A6801, A6806, A6812	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A6801, A6806, A6812	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A6801, A6806, A6809, A6812	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A6801, A6804, A6805, A6806, A6807, A6808, A6809, A6810, A6811, A6812, A6813, A6814	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A6801	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A6802	Component - No Curve - P-256, P-384 Hash Algorithm - SHA-1	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A6802	Curve - P-256, P-384 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-256, SHA3-384, SHA3-512	FIPS 186-5
Hash DRBG	A6801, A6815, A6816, A6817	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
HMAC DRBG	A6801, A6815, A6816, A6817	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-90A Rev. 1
HMAC-SHA-1	A6801, A6815, A6816, A6817, A6818	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A6801, A6815, A6816, A6817, A6818	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A6801, A6815, A6816, A6817, A6818	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A6801, A6815, A6816, A6817	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A6801, A6815, A6816, A6817	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A6801	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A6801	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A6801	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A6801	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6801	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A6801	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA OneStep SP800-56Cr2	A6803	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2
KDF SP800-108	A6803	KDF Mode - Counter Supported Lengths - Supported Lengths: 112-4096 Increment 8	SP 800-108 Rev. 1
RSA SigVer (FIPS186-4)	A6801	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A6801	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
Safe Primes Key Generation	A6801	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	SP 800-56A Rev. 3
SHA-1	A6801, A6815, A6816, A6817, A6818	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-224	A6801, A6815, A6816, A6817, A6818	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A6801, A6815, A6816, A6817, A6818	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-384	A6801, A6815, A6816, A6817	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-512	A6801, A6815, A6816, A6817	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA3-224	A6801	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-256	A6801	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-384	A6801	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-512	A6801	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG (asymmetric)	Key Type:Asymmetric	N/A	SP800-133r2, section 4 example 1 (without XOR)

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 with external IV (not compliant to FIPS 140-3 IG C.H)
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

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Encryption and Decryption with AES	BC-UnAuth	SP800-38A. Encryption, Decryption		AES-CBC: (A6801, A6806, A6809, A6812) AES-ECB: (A6801, A6804, A6805, A6806, A6807, A6808, A6809, A6810, A6811, A6812, A6813, A6814) AES-CFB128: (A6801, A6806, A6812) AES-XTS Testing Revision 2.0: (A6801, A6806, A6809, A6812) AES-CBC-CS3: (A6801, A6806, A6809, A6812) AES-OFB: (A6801, A6806, A6812)
Authenticated Encryption and Authenticated Decryption with AES-KW	BC-Auth	SP800-38F. Authenticated encryption, Authenticated decryption	Authenticated Encryption and Authenticated Decryption with AES-KW Security Strength:128-256 bits	AES-KW: (A6801, A6806, A6812)
Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG	DRBG	SP800-90Ar1. Random number generation		Hash DRBG: (A6801, A6815, A6816, A6817) HMAC DRBG: (A6801, A6815, A6816, A6817) Counter DRBG: (A6801, A6804, A6805, A6806, A6807, A6808, A6809, A6810, A6811, A6812, A6813, A6814)
Message Authentication Code Generation with AES or HMAC	MAC	SP800-38B, SP800-38D, FIPS 198-1. Message authentication		HMAC-SHA-1: (A6801, A6815, A6816, A6817, A6818) HMAC-SHA2-224: (A6801, A6815, A6816, A6817, A6818) HMAC-SHA2-256:

Name	Type	Description	Properties	Algorithms
				(A6801, A6815, A6816, A6817, A6818) HMAC-SHA2-384: (A6801, A6815, A6816, A6817) HMAC-SHA2-512: (A6801, A6815, A6816, A6817) AES-CMAC: (A6801, A6806, A6812) AES-GMAC: (A6801, A6806, A6812) HMAC-SHA3-224: (A6801) HMAC-SHA3-256: (A6801) HMAC-SHA3-384: (A6801) HMAC-SHA3-512: (A6801)
Message Authentication Code Verification with AES-GMAC	MAC	SP800-38D, FIPS 198-1. Message authentication		AES-GMAC: (A6801, A6806, A6812)
Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC	KAS-SSC	SP 800-56Ar3. KAS-ECC-SSC and KAS-FFC-SSC per IG D.F Scenario 2 (1)	KAS-FFC-SSC Security Strength:112-200 bits KAS-ECC-SSC Security Strength:128, 192 bits	KAS-ECC-SSC Sp800-56Ar3: (A6801) KAS-FFC-SSC Sp800-56Ar3: (A6801)
Message Digest with SHA	SHA	FIPS180-4, FIPS202. Message digest		SHA-1: (A6801, A6815, A6816, A6817, A6818) SHA2-224: (A6801, A6815, A6816, A6817, A6818) SHA2-256: (A6801, A6815, A6816, A6817, A6818) SHA2-384: (A6801, A6815, A6816, A6817) SHA2-512: (A6801, A6815, A6816, A6817) SHA3-224: (A6801)

Name	Type	Description	Properties	Algorithms
				SHA3-256: (A6801) SHA3-384: (A6801) SHA3-512: (A6801)
Key Pair Generation with ECDSA or Safe Primes	AsymKeyPair-KeyGen CKG KAS-KeyGen	FIPS186-5, SP800-56Ar3. ECDSA Key pair generation according to FIPS186-5, Appendix A.2.2 per IG D.H and SP800-133r2, section 4 (without XOR) 5.1, 5.2; Safe Primes Key Generation according to SP800-56Ar3, Section 5.6.1.1.4 per IG D.H and SP800-133r2, section 4 (without XOR), 5.2		Safe Primes Key Generation: (A6801) ECDSA KeyGen (FIPS186-5): (A6801) CKG (asymmetric): ()
Authenticated Encryption and Authenticated Decryption with AES-CCM	BC-Auth	SP800-38C. Authenticated encryption, Authenticated decryption	Authenticated Encryption and Authenticated Decryption with AES-CCM Security Strength:128-256 bits	AES-CCM: (A6801, A6806, A6812)
Authenticated Encryption and Authenticated Decryption with AES-GCM	BC-Auth	SP800-38D. Authenticated encryption, Authenticated decryption	Authenticated Encryption and Authenticated Decryption with AES-GCM Security Strength:128-256 bits	AES-GCM: (A6801, A6804, A6805, A6806, A6807, A6808, A6809, A6810, A6811, A6812, A6813, A6814)
Authenticated Encryption and Authenticated Decryption with AES-CBC or AES-CTR with HMAC	BC-Auth	SP800-38A, FIPS 198-1. Authenticated encryption, Authenticated decryption		AES-CBC: (A6801, A6806, A6809, A6812) AES-CTR: (A6801, A6806, A6809, A6812) HMAC-SHA-1: (A6801, A6815, A6816, A6817, A6818) HMAC-SHA2-256: (A6801, A6815, A6816, A6817,

Name	Type	Description	Properties	Algorithms
				A6818) HMAC-SHA2-384: (A6801, A6815, A6816, A6817) HMAC-SHA2-512: (A6801, A6815, A6816, A6817)
Signature Verification with RSA	DigSig-SigVer	Signature verification		RSA SigVer (FIPS186-5): (A6801) SHA2-224: (A6801, A6815, A6816, A6817, A6818) SHA2-256: (A6801, A6815, A6816, A6818) SHA2-384: (A6801, A6815, A6816, A6817) SHA2-512: (A6801, A6815, A6816, A6817)
Legacy Signature Verification with RSA	DigSig-SigVer	Signature verification		SHA-1: (A6801, A6815, A6816, A6817, A6818) RSA SigVer (FIPS186-4): (A6801)
Signature Verification with ECDSA	DigSig-SigVer	Signature verification		SHA2-224: (A6801, A6815, A6816, A6817, A6818) SHA2-256: (A6801, A6815, A6816, A6817, A6818) SHA2-384: (A6801, A6815, A6816, A6817) SHA2-512: (A6801, A6815, A6816, A6817) SHA3-256: (A6801) SHA3-384: (A6801) SHA3-512: (A6801) ECDSA SigVer (FIPS186-5): (A6802)

Name	Type	Description	Properties	Algorithms
Legacy Signature Verification with ECDSA	DigSig-SigVer	Signature verification		SHA-1: (A6801, A6815, A6816, A6817, A6818) ECDSA SigVer (FIPS186-4): (A6802)
Key Derivation with KBKDF	KBKDF	SP 800-108r1. Key derivation		KDF SP800-108: (A6803)
Key Derivation with KDA OneStep	KAS-56CKDF	SP 800-56cr2. Key agreement scheme		KDA OneStep SP800-56Cr2: (A6803)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

Algorithms designated as “Legacy” can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

2.7.1 AES GCM IV

For IPsec, the module offers the AES GCM implementation and uses the context of Scenario 1 (b) 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 the *Approved Services* table.

2.7.2 AES XTS

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

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

To meet the requirement stated in IG C.I, the module implements a check to ensure that the two AES keys used in AES XTS mode are not identical. As the module does not implement symmetric key generation, this check is performed when the keys are input by the operator. Key_1 and Key_2 shall be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133r2, Section 6.3.

2.7.3 Diffie-Hellman and EC Diffie-Hellman

The module offers DH and ECDH shared secret computation services compliant to the SP 800-56Ar3 and meeting IG D.F scenario 2 path (1). To meet the required assurances listed in Section 5.6 of SP 800-56Ar3, the module shall be used together with an application that implements the IPSec protocol and the following steps shall be performed:

1. The entity using the module, must use the module's "Key pair generation" service: the set_secret and generate_public_key API functions, to generate DH/ECDH ephemeral key pairs. This meets the assurances required by key pair owner defined in the section 5.6.2.1 of SP 800-56Ar3.

2. As part of the module's shared secret computation service, the module internally performs the public key validation on the peer's public key passed in as input to the API function. This meets the public key validity assurance required by section 5.6.2.2.2 of SP 800-56Ar3.

3. The module does not support static keys, therefore the "assurance of peer's possession of private key" is not applicable.

2.7.5 SHA-1

Digital signature generation using SHA-1 is non-approved and not allowed in approved services.

2.7.6 SHA-3

The module implements HMAC with SHA3-224, SHA3-256, SHA3-384, SHA3-512. The CAVP certificates have been obtained for the HMAC algorithm as well as for all the SHA3 implementations. The CAVP certificates are listed in the *Approved Algorithms* table.

2.7.7 RSA

The module implements FIPS 186-4 RSA SigVer and FIPS 186-5 RSA SigVer. All RSA modulus lengths (i.e., 2048, 3072, 4096 bits) have been CAVP tested. The CAVP certificates are listed in the *Approved Algorithms* table.

2.8 RBG and Entropy

Cert Number	Vendor Name
E260	IBM Corporation

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
IBM Corporation Kernel CPU Time Jitter Entropy source Version 3.4.0	Non-Physical	ClevOS 3.19 on Intel® Xeon® 8474C	256 bits	256 bits	SHA3-256 (A6801)

Table 11: Entropy Sources

The module implements three different Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90Ar1: Counter DRBG, Hash DRBG, and HMAC DRBG. Each of these DRBG implementations can be instantiated by the operator of the module, using the parameters listed specified in the *Security Function Implementations* table. 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 module complies with the Public Use Document for ESV certificate E260 by reading entropy data from the jent_kcapi_random() function, which corresponds to the GetEntropy() conceptual interface. This function outputs 256 bits of full entropy.

The HMAC-SHA-512 DRBG is instantiated with a 384-bit entropy input and reseeded with a 256-bits long 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 operational environment on the ESV certificate is identical to the operating system described in this document, and the entropy source is implemented inside the cryptographic boundary. Thus, the module is compliant with scenario 1 of IG 9.3.A. There are no maintenance requirements for the entropy source.

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 directly obtained as output from the SP 800-90Ar1 approved DRBG, compliant with Section 4 of SP 800-133r2 (without XOR, as described in Additional Comment 2 of IG D.H). The following methods are implemented:

- 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.
- ECDSA key pair generation: compliant with SP 800-133r2, Section 5.1 and 5.2. The method described in Appendix A.2.2 of FIPS 186-5 (“Rejection Sampling”) is used. Note that this generation method is also used to generated ECDH key pairs.

Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

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.

The module implements shared secret computation for DH and ECDH following SP 800-56Arev3

No other parts of the TLS 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

N/A for this module.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
CO	Role	CO	None

Table 13: 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.

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 with SHA	CO
Encryption	Encrypt a plaintext	crypto_skcipher_setkey returns 0	AES Key, plaintext	Ciphertext	Encryption and Decryption with AES	CO - AES Key: W,E
Decryption	Decrypt a ciphertext	crypto_skcipher_setkey returns 0	AES Key, ciphertext	Plaintext	Encryption and Decryption with AES	CO - 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 and Authenticated Decryption with AES-CCM Authenticated Encryption and Authenticated Decryption with AES-GCM Authentica	CO - AES Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ted Encryption and Authenticated Decryption with AES-CBC or AES-CTR with HMAC Authenticated Encryption and Authenticated Decryption with AES-KW	
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, MAC tag, ciphertext	Plaintext or failure	Authenticated Encryption and Authenticated Decryption with AES-CCM Authenticated Encryption and Authenticated Decryption with AES-GCM Authenticated Encryption and Authenticated Decryption with AES-CBC or AES-CTR with HMAC Authenticated Encryption and Authentica	CO - AES Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ted Decryption with AES-KW	
Message Authentication Code Generation	Compute a MAC tag	crypto_shash_init returns 0	AES key or HMAC key, message	MAC tag	Message Authentication Code Generation with AES or HMAC	CO - AES Key: W,E - HMAC Key: W,E
Message Authentication Code Verification	Verify a MAC tag	crypto_shash_init returns 0	AES key, MAC tag, message	Pass/fail	Message Authentication Code Verification with AES-GMAC	CO - AES Key: W,E
Random Number Generation	Generate random bytes	crypto_rng_get_bytes returns 0	Output length	Random bytes	Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG	CO - Entropy Input (IG D.L): W,E,Z - CTR_DRBG Seed (IG D.L): G,E - Hash_DRBG Seed (IG D.L): G,E - HMAC_DRBG Seed (IG D.L): G,E - CTR_DRBG Internal State (V, Key) (IG D.L): G,W,E - Hash_DRBG Internal State (V, C) (IG D.L): G,W,E - HMAC_DR

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						BG Internal State (V, Key) (IG D.L): G,W,E
Shared Secret Computation	Compute a shared secret	crypto_kpp_compute_shared_secret returns 0	DH private key, DH public key or EC private key, EC public key	Shared secret	Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC	CO - DH Public Key: W,E - DH Private Key: W,E - EC Public Key: W,E - EC Private Key: W,E - Shared Secret: G,R
Key Pair Generation	Generate a key pair	crypto_kpp_set_secret and crypto_kpp_generate_public_key return 0	Safe Primes: Group; ECDSA: Curve	Safe Primes: DH private key, DH public key; ECDSA: EC private key, EC public key	Key Pair Generation with ECDSA or Safe Primes	CO - Intermediate Key Generation Value: G,E,Z - DH Public Key: G,R - DH Private Key: G,R - EC Public Key: G,R - EC Private Key: G,R
Signature Verification	Verify a signature	crypto_akcipher_init returns 0	Signature, ECDSA public key, RSA public key	Digital signature verification result	Signature Verification with RSA Signature Verification with ECDSA Legacy Signature Verification with RSA Legacy	CO - RSA Public Key: W,E - EC Public Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Signature Verification with ECDSA	
KBKDF Key Derivation	Derive a key from a key-derivation key	crypto_kdf108_ctr_generate returns 0	Key-Derivation Key	KBKDF Derived Key	Key Derivation with KBKDF	CO - Key-Derivation Key: W,E - KBKDF Derived Key: G,R
OneStep KDA Key Derivation	Derive a key from a shared secret	crypto_kdf108_ctr_generate returns 0	Shared Secret	KDA OneStep Derived Key	Key Derivation with KDA OneStep	CO - Shared Secret: W,E - KDA OneStep Derived Key: G,R
Error Detection Code	Compute an EDC (crc32, crct10dif, crc64-rocksoft)	None	Message	EDC	None	CO
Compression	Compress data (deflate, deflate-iaa, lz4, lz4hc, lzo, zstd)	None	Data	Compressed data	None	CO
Generic System Call	Use the kernel to perform various non-cryptographic operations	None	Identifier, various arguments	Various return values	None	CO
Show Version	Return the module name and version information	None	N/A	Module name and version	None	CO
Show Status	Return the module status	None	N/A	Module status	None	CO
Self-Test	Perform the CASTs and	None	N/A	Pass/fail	Encryption and Decryption	CO

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	integrity tests				with AES Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG Message Authentication Code Generation with AES or HMAC Message Authentication Code Verification with AES-GMAC Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC Message Digest with SHA Key Pair Generation with ECDSA or Safe Primes Authenticated Encryption and Authenticated Decryption with AES-CCM Authenticated Encryption and Authentica	

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ted Decryption with AES-GCM Authenticated Encryption and Authenticated Decryption with AES-CBC or AES-CTR with HMAC Authenticated Encryption and Authenticated Decryption with AES-KW Signature Verification with RSA Signature Verification with ECDSA	
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	None	CO - AES Key: Z - HMAC Key: Z - Shared Secret: Z - Entropy Input (IG D.L): Z - CTR_DRBG Seed (IG D.L): Z - Hash_DRBG Seed (IG D.L): Z - HMAC_DR

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						BG Seed (IG D.L): Z - CTR_DRBG Internal State (V, Key) (IG D.L): Z - Hash_DRBG Internal State (V, C) (IG D.L): Z - HMAC_DRBG Internal State (V, Key) (IG D.L): Z - DH Public Key: Z - DH Private Key: Z - EC Public Key: Z - EC Private Key: Z - Intermediate Key Generation Value: Z - RSA Public Key: Z - Key-Derivation Key: Z - KBKDF Derived Key: Z - KDA OneStep

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Derived Key: Z

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

Not applicable.

5 Software/Firmware Security

5.1 Integrity Techniques

The kcapi-hasher binary utilizes the module's HMAC and SHA-512 implementations and verifies it's integrity test and the libkcapi library integrity followed by the integrity test on the static kernel binary i.e. vmlinuz. The HMAC key used for this integrity test is stored in libkcapi. The kernel object (.ko) files are verified using RSA signature verification with PKCS#1 v1.5 padding, SHA-512, and a 4096-bit key stored in the kernel.

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 module executes as part of a general-purpose operating system (ClevOS 3.19), which allows modification, loading, and execution of software that is not part of the validated module.

The approved cryptographic algorithms of the module are part of the Linux kernel, which operates in Linux kernel space. This ensures that any SSPs contained within the module are protected by the process isolation and memory separation mechanisms provided by the Linux kernel, and only the module has control over these SSPs. The user space libkcapi and kcapi-hasher components, though not processing any SSPs, are similarly protected by the operating environment.

6.2 Configuration Settings and Restrictions

The module shall be installed as specified 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 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; 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 17: 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	By calling the appropriate zeroization functions: AES key: crypto_free_skcipher and crypto_free_aead; HMAC key: crypto_free_shash and crypto_free_ahash; Internal state: crypto_free_rng; DH public & private key: crypto_free_kpp; EC public & private key: crypto_free_kpp; Key-Derivation Key: memzero_explicit; KBKDF Derived Key: memzero_explicit; KDA OneStep Derived Key: memzero_explicit
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which	N/A

Zeroization Method	Description	Rationale	Operator Initiation
		renders the SSP values irretrievable.	
Remove power from the module	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By removing power

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 used for Encryption; Decryption; Authenticated encryption; Authenticated decryption; Message authentication;	XTS: 128, 256 bits; ECB, CBC, CTR, CFB128, CBC-CS3, KW, OFB, CCM, GCM, CMAC, GMAC: 128, 192, 256 bits - XTS: 128, 256 bits; ECB, CBC, CTR, CFB128, CBC-CS3, KW, OFB, CCM, GCM, CMAC, GMAC: 128, 192, 256 bits	Symmetric key - CSP			Encryption and Decryption with AES Message Authentication Code Generation with AES or HMAC Message Authentication Code Verification with AES-GMAC
HMAC Key	HMAC key used for Message authentication code (MAC);	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Code Generation with AES or HMAC
Shared Secret	Shared secret established during Shared Secret Computation	KAS-FFC-SSC:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192; KAS-ECC-SSC: P-256, P-384 bits - KAS-	Shared secret - CSP		Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC	Key Derivation with KDA OneStep

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		FFC-SSC: 112-200 bits; KAS-ECC-SSC: 128, 192 bits				
Entropy Input (IG D.L)	Entropy input used to seed the DRBGs	128-384 bits - 128-384 bits	Entropy input - CSP			Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
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 with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
Hash_DRBG Seed (IG D.L)	DRBG seed derived from Entropy Input	440, 888 bits - 128, 256 bits	Seed - CSP	Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
HMAC_DRBG Seed (IG D.L)	DRBG seed derived from Entropy Input	440, 888 bits - 128, 256 bits	Seed - CSP	Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
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 with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
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 with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
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 with HMAC DRBG, Hash DRBG or Counter DRBG		Random Number Generation with HMAC DRBG, Hash DRBG or Counter DRBG
DH Public Key	Public key used for KAS-FFC-SSC	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Public key - PSP	Key Pair Generation with ECDSA or Safe Primes		Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC
DH Private Key	DH private key used for KAS-FFC-SSC	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Private key - CSP	Key Pair Generation with ECDSA or Safe Primes		Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC
EC Public Key	Public key used for KAS-ECC-SSC	P-256, P-384 - 128, 192 bits	Public key - PSP	Key Pair Generation with ECDSA or Safe Primes		Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC Signature Verification with ECDSA
EC Private Key	EC private key used for KAS-ECC-SSC	P-521, P-384 - 128, 192 bits	Private key - CSP	Key Pair Generation with ECDSA or Safe Primes		Shared Secret Computation with KAS-FFC-SSC or KAS-ECC-SSC
Intermediate Key Generation Value	Intermediate value generated during Key Pair Generation	192-8192 bits - 112-256 bits	Intermediate value - CSP	Key Pair Generation with ECDSA or Safe Primes		Key Pair Generation with ECDSA or Safe Primes

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA Public Key	RSA Public key used for Signature Verification with RSA	2048-4096 bits - 112-150 bits	Public key - PSP			Signature Verification with RSA
Key-Derivation Key	Used for key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key Derivation with KBKDF
KBKDF Derived Key	Generated by key-based key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KBKDF		
KDA OneStep Derived Key	Generated by OneStep KDA key derivation	2048 bits - 112 bits	Symmetric key - CSP	Key Derivation with KDA OneStep		

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	
HMAC Key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	
Shared Secret	API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	DH Public Key:Derived From DH Private Key:Derived From EC Public Key:Derived From EC Private Key:Derived From
Entropy Input (IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic Remove power from the module	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 Remove power from the module	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	Automatic Remove	Entropy Input (IG D.L):Derived From Hash_DRBG

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			service completion	power from the module	Internal State (V, C) (IG D.L):Derives
HMAC_DRBG Seed (IG D.L)		RAM:Plaintext	From service invocation to service completion	Automatic Remove power from the module	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 service completion	Free cipher handle Remove power from the module	CTR_DRBG Seed (IG D.L):Derived From
Hash_DRBG Internal State (V, C) (IG D.L)		RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	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	Free cipher handle Remove power from the module	HMAC_DRBG Seed (IG D.L):Derived From
DH Public Key	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	DH Private Key:Paired With Shared Secret:Derives Intermediate Key Generation Value:Generated From
DH Private Key	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	DH Public Key:Paired With Shared Secret:Derives Intermediate Key Generation Value:Generated From
EC Public Key	API input parameters; AF_ALG_type sockets (input) API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	EC Private Key:Paired With Shared Secret:Derives Intermediate Key Generation Value:Generated From
EC Private Key	API input parameters; AF_ALG_type	RAM:Plaintext	From service invocation to	Free cipher handle Remove	EC Public Key:Paired With Shared

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	sockets (input) API output parameters; AF_ALG type sockets (output)		service completion	power from the module	Secret:Derives Intermediate Key Generation Value:Generated From
Intermediate Key Generation Value		RAM:Plaintext	From service invocation to service completion	Automatic	DH Public Key:Generates DH Private Key:Generates EC Public Key:Generates EC Private Key:Generates
RSA Public Key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Automatic	
Key-Derivation Key	API input parameters; AF_ALG_type sockets (input)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	KBKDF Derived Key:Derives
KBKDF Derived Key	API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	Key-Derivation Key:Derived From
KDA OneStep Derived Key	API output parameters; AF_ALG type sockets (output)	RAM:Plaintext	From service invocation to service completion	Free cipher handle Remove power from the module	Shared Secret:Derived From

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 January 1, 2031.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-512 (A6817) - kernel	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Used for kernel binary
HMAC-SHA2-512 (A6817) - libkcapi	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Used for libkcapi shared library
HMAC-SHA2-512 (A6817) - kcapi-hasher	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Used for kcapi-hasher binary
RSA SigVer (FIPS186-5) (A6801)	4096-bit key with SHA-512	Signature Verification	SW/FW Integrity	Module becomes operational and services are available for use	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-ECB (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6804) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6805) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A6807) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6808) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6809) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6810) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6811) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6813) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6814) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC (A6809) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC-CS3 (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6809) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CTR (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CTR (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CTR (A6809) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CTR (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CCM (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CCM (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-CCM (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6801) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6804) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6805) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A6806) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6807) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6808) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6809) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6810) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6811) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6812) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6813) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-GCM (A6814) - Encrypt	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6801) - Encrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6806) - Encrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6809) - Encrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-XTS Testing Revision 2.0 (A6812) - Encrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at power-on before the integrity test
AES-ECB (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6804) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6805) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6807) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6808) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6809) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6810) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6811) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6813) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-ECB (A6814) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
AES-CBC (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC (A6809) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6809) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CBC-CS3 (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CTR (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CTR (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CTR (A6809) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CTR (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CCM (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CCM (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CCM (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6801) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6804) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6805) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6806) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6807) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6808) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6809) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6810) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6811) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6812) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A6813) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-GCM (A6814) - Decrypt	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6801) - Decrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6806) - Decrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6809) - Decrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-XTS Testing Revision 2.0 (A6812) - Decrypt	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at power-on before the integrity test
AES-CMAC (A6801)	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A6806)	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
AES-CMAC (A6812)	128-bit and 256-bit key	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
SHA-1 (A6801)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A6815)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A6816)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A6817)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A6818)	SHA-1	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6801)	SHA2-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6815)	SHA2-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6816)	SHA2-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6817)	SHA2-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-224 (A6818)	SHA2-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A6801)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A6815)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A6816)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A6817)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-256 (A6818)	SHA2-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A6801)	SHA2-384	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
SHA2-384 (A6815)	SHA2-384	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A6816)	SHA2-384	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-384 (A6817)	SHA2-384	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A6801)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A6815)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A6816)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A6817)	SHA2-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-224 (A6801)	SHA3-224	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A6801)	SHA3-256	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-384 (A6801)	SHA3-384	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-512 (A6801)	SHA3-512	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
HMAC-SHA-1 (A6801)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A6815)	SHA-1	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-SHA-1 (A6816)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A6817)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1 (A6818)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A6801)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A6815)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A6816)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A6817)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-224 (A6818)	SHA2-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A6801)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A6815)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A6816)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A6817)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A6818)	SHA2-256	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-SHA2-384 (A6801)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A6815)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A6816)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-384 (A6817)	SHA2-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A6801)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A6815)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A6816)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512 (A6817)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-224 (A6801)	SHA3-224	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-256 (A6801)	SHA3-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-384 (A6801)	SHA3-384	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA3-512 (A6801)	SHA3-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
Counter DRBG (A6801)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed,	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
					generate) health test	
Counter DRBG (A6804)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6805)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6806)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6807)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6808)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6809)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6810)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6811)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6812)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Counter DRBG (A6813)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	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
Counter DRBG (A6814)	128, 192, 256 bit keys, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Hash DRBG (A6801)	SHA2-256 With/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Hash DRBG (A6815)	SHA2-256 With/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Hash DRBG (A6816)	SHA2-256 With/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
Hash DRBG (A6817)	SHA2-256 With/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A6801)	HMAC-SHA2-256, HMAC-SHA2-512, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A6815)	HMAC-SHA2-256, HMAC-SHA2-512, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A6816)	HMAC-SHA2-256, HMAC-SHA2-512, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
HMAC DRBG (A6817)	HMAC-SHA2-256, HMAC-SHA2-512, with/without PR	KAT	CAST	Module becomes operational	SP 800-90Ar1 (instantiate, reseed, generate) health test	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A6801)	P-256, P-384 curves	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A6801)	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-4) (A6801)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A6801)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A6802)	SHA2-256, P-256 curve	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A6802)	SHA2-256, P-256 curve	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
KDF SP800-108 (A6803)	Counter mode; HMAC-SHA-256; 256-bit input key	KAT	CAST	Module becomes operational	Key based key derivation	Test runs at power-on before the integrity test
Safe Primes Key Generation (A6801)	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, Section 5.6.1.1.4 Testing Candidates	PCT	PCT	Successful key pair generation	SP 800-56ARev3, 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5) (A6801)	SHA2-256, P-256, P-384 curves, Appendix A.2.2 Rejection Sampling	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
KDA OneStep SP800-56Cr2 (A6803)	SHA2-256	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
Entropy Source Initialization RCT	1024 samples	RCT	CAST	Module becomes operational and services are available for use	Entropy source startup test	Entropy source initialization
Entropy Source Initialization APT	1024 samples	APT	CAST	Module becomes operational and services are available for use	Entropy source startup test	Entropy source initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Entropy Source Operational RCT	Intermittent Cutoff: 31 samples, Permanent Cutoff: 61 samples	RCT	CAST	Entropy source is operational	Entropy source continuous test	Continuously
Entropy Source Operational APT	512 samples, Intermittent Cutoff: 325 samples, Permanent Cutoff: 355 samples	APT	CAST	Entropy source is operational	Entropy source continuous test	Continuously

Table 22: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A6817) - kernel	Message Authentication	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-512 (A6817) - libkcapi	Message Authentication	SW/FW Integrity	On Demand	Manually
HMAC-SHA2-512 (A6817) - kcapi-hasher	Message Authentication	SW/FW Integrity	On Demand	Manually
RSA SigVer (FIPS186-5) (A6801)	Signature Verification	SW/FW Integrity	On Demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6804) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6805) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6807) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6808) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6809) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6810) - Encrypt	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A6811) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6813) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6814) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6809) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6809) - Encrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6809) - Encrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6804) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6805) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6807) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6808) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6809) - Encrypt	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A6810) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6811) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6813) - Encrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6814) - Encrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6801) - Encrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6806) - Encrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6809) - Encrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6812) - Encrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6804) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6805) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6807) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6808) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6809) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6810) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6811) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6813) - Decrypt	KAT	CAST	On Demand	Manually
AES-ECB (A6814) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC (A6809) - Decrypt	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6809) - Decrypt	KAT	CAST	On Demand	Manually
AES-CBC-CS3 (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6809) - Decrypt	KAT	CAST	On Demand	Manually
AES-CTR (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-CCM (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6801) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6804) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6805) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6807) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6808) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6809) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6810) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6811) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6813) - Decrypt	KAT	CAST	On Demand	Manually
AES-GCM (A6814) - Decrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6801) - Decrypt	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-XTS Testing Revision 2.0 (A6806) - Decrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6809) - Decrypt	KAT	CAST	On Demand	Manually
AES-XTS Testing Revision 2.0 (A6812) - Decrypt	KAT	CAST	On Demand	Manually
AES-CMAC (A6801)	KAT	CAST	On Demand	Manually
AES-CMAC (A6806)	KAT	CAST	On Demand	Manually
AES-CMAC (A6812)	KAT	CAST	On Demand	Manually
SHA-1 (A6801)	KAT	CAST	On Demand	Manually
SHA-1 (A6815)	KAT	CAST	On Demand	Manually
SHA-1 (A6816)	KAT	CAST	On Demand	Manually
SHA-1 (A6817)	KAT	CAST	On Demand	Manually
SHA-1 (A6818)	KAT	CAST	On Demand	Manually
SHA2-224 (A6801)	KAT	CAST	On Demand	Manually
SHA2-224 (A6815)	KAT	CAST	On Demand	Manually
SHA2-224 (A6816)	KAT	CAST	On Demand	Manually
SHA2-224 (A6817)	KAT	CAST	On Demand	Manually
SHA2-224 (A6818)	KAT	CAST	On Demand	Manually
SHA2-256 (A6801)	KAT	CAST	On Demand	Manually
SHA2-256 (A6815)	KAT	CAST	On Demand	Manually
SHA2-256 (A6816)	KAT	CAST	On Demand	Manually
SHA2-256 (A6817)	KAT	CAST	On Demand	Manually
SHA2-256 (A6818)	KAT	CAST	On Demand	Manually
SHA2-384 (A6801)	KAT	CAST	On Demand	Manually
SHA2-384 (A6815)	KAT	CAST	On Demand	Manually
SHA2-384 (A6816)	KAT	CAST	On Demand	Manually
SHA2-384 (A6817)	KAT	CAST	On Demand	Manually
SHA2-512 (A6801)	KAT	CAST	On Demand	Manually
SHA2-512 (A6815)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-512 (A6816)	KAT	CAST	On Demand	Manually
SHA2-512 (A6817)	KAT	CAST	On Demand	Manually
SHA3-224 (A6801)	KAT	CAST	On Demand	Manually
SHA3-256 (A6801)	KAT	CAST	On Demand	Manually
SHA3-384 (A6801)	KAT	CAST	On Demand	Manually
SHA3-512 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A6815)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A6816)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A6817)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A6818)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A6815)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A6816)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A6817)	KAT	CAST	On Demand	Manually
HMAC-SHA2-224 (A6818)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6815)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6816)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6817)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A6818)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A6815)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A6816)	KAT	CAST	On Demand	Manually
HMAC-SHA2-384 (A6817)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A6801)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A6815)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A6816)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A6817)	KAT	CAST	On Demand	Manually
HMAC-SHA3-224 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA3-256 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA3-384 (A6801)	KAT	CAST	On Demand	Manually
HMAC-SHA3-512 (A6801)	KAT	CAST	On Demand	Manually
Counter DRBG (A6801)	KAT	CAST	On Demand	Manually
Counter DRBG (A6804)	KAT	CAST	On Demand	Manually
Counter DRBG (A6805)	KAT	CAST	On Demand	Manually
Counter DRBG (A6806)	KAT	CAST	On Demand	Manually
Counter DRBG (A6807)	KAT	CAST	On Demand	Manually
Counter DRBG (A6808)	KAT	CAST	On Demand	Manually
Counter DRBG (A6809)	KAT	CAST	On Demand	Manually
Counter DRBG (A6810)	KAT	CAST	On Demand	Manually
Counter DRBG (A6811)	KAT	CAST	On Demand	Manually
Counter DRBG (A6812)	KAT	CAST	On Demand	Manually
Counter DRBG (A6813)	KAT	CAST	On Demand	Manually
Counter DRBG (A6814)	KAT	CAST	On Demand	Manually
Hash DRBG (A6801)	KAT	CAST	On Demand	Manually
Hash DRBG (A6815)	KAT	CAST	On Demand	Manually
Hash DRBG (A6816)	KAT	CAST	On Demand	Manually
Hash DRBG (A6817)	KAT	CAST	On Demand	Manually
HMAC DRBG (A6801)	KAT	CAST	On Demand	Manually
HMAC DRBG (A6815)	KAT	CAST	On Demand	Manually
HMAC DRBG (A6816)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC DRBG (A6817)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A6801)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A6801)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A6801)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A6801)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A6802)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A6802)	KAT	CAST	On Demand	Manually
KDF SP800-108 (A6803)	KAT	CAST	On Demand	Manually
Safe Primes Key Generation (A6801)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A6801)	PCT	PCT	On Demand	Manually
KDA OneStep SP800-56Cr2 (A6803)	KAT	CAST	On Demand	Manually
Entropy Source Initialization RCT	RCT	CAST	On Demand	Manually
Entropy Source Initialization APT	APT	CAST	On Demand	Manually
Entropy Source Operational RCT	RCT	CAST	On Demand	Manually
Entropy Source Operational APT	APT	CAST	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

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

On the ClevOS 3.19 operational environment, the module is distributed within the clevos-3.19.2.F1606-44_fips-accesser-usbiso.iso image.

There are no specific steps for installing the module, the module is installed as part of operating system.

11.2 Administrator Guidance

The Approved and non-Approved modes of operation are specified in section 2.4. The administrative functions are specified in the *Approved Services* table. All the physical ports and logical interfaces are specified in section 3.1.

The Crypto Officer must execute the Show version service using following commands:

$ cat /proc/sys/crypto/fips_name

The Crypto Officer must ensure that the proper name is listed in the output as follows:

IBM COS Linux Kernel Cryptographic API

Then, the Crypto Officer must execute:

$ cat /proc/sys/crypto/fips_version

This command must output the following version for kernel components:

3.0

Then, the Crypto Officer must execute:

$ apt list --installed | grep libkcapi

This command must output the following version for libkcapi and kcapi-hasher components:

1.5.0-1 amd64

On the ClevOS 3.19 operational environments, versions of the installed packages can be verified using the following command:

$ dpkg-query -W linux-image-6.1.0-32-amd64 libkcapi1 kcapi-tools

11.3 Non-Administrator Guidance

The approved and non-approved security functions available to users are listed in section 2, the physical ports, and logical interfaces available to users are specified in section 3.1. The Approved and non-Approved modes of operation are specified in section 2.4. The algorithm-specific information is listed in section 2.7.

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.

If desired, the linux-image-6.1.0-32-amd64, libkcapi1, and kcapi-tools deb packages can be uninstalled from the ClevOS 3.19 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
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IPsec	Internet Protocol Security
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
KW	Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PAI	Processor Algorithm Implementation
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS 140-3	**FIPS PUB 140-3 - Security Requirements For Cryptographic Modules** March 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG (Last Update: April 18, 2025)	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	**Secure Hash Standard (SHS)** March 2012 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS 186-4	**Digital Signature Standard (DSS)** July 2013 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS 186-5	**Digital Signature Standard (DSS)** February 2023 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf
FIPS 197	**Advanced Encryption Standard** May 9, 2023 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf
FIPS 198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1** February 2003 https://www.ietf.org/rfc/rfc3447.txt
RFC 3526	**More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)** May 2003 https://www.ietf.org/rfc/rfc3526.txt
RFC 4106	**The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)** June 2005 https://datatracker.ietf.org/doc/html/rfc4106
RFC 7296	**Internet Key Exchange Protocol Version 2 (IKEv2)** October 2014 https://datatracker.ietf.org/doc/html/rfc7296
SP 800-38A	**Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

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