NetApp, Inc.

NetApp StorageGRID Kernel Crypto API

FIPS 140-3 Non-Proprietary Security Policy

Prepared by:

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

Document version: 1.1
Last update: 2025-11-25

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 kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0 of the NetApp StorageGRID Kernel Cryptographic API 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.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

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

Module Type: Software

Module Embodiment: MultiChipStand

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the kernel binary, the libkcapi shared library, and the sha512hmac binary, which is used to verify the integrity of the software components. In addition, the cryptographic boundary contains the .hmac files which store the expected integrity values for each of the software components. The cryptographic boundary is 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.

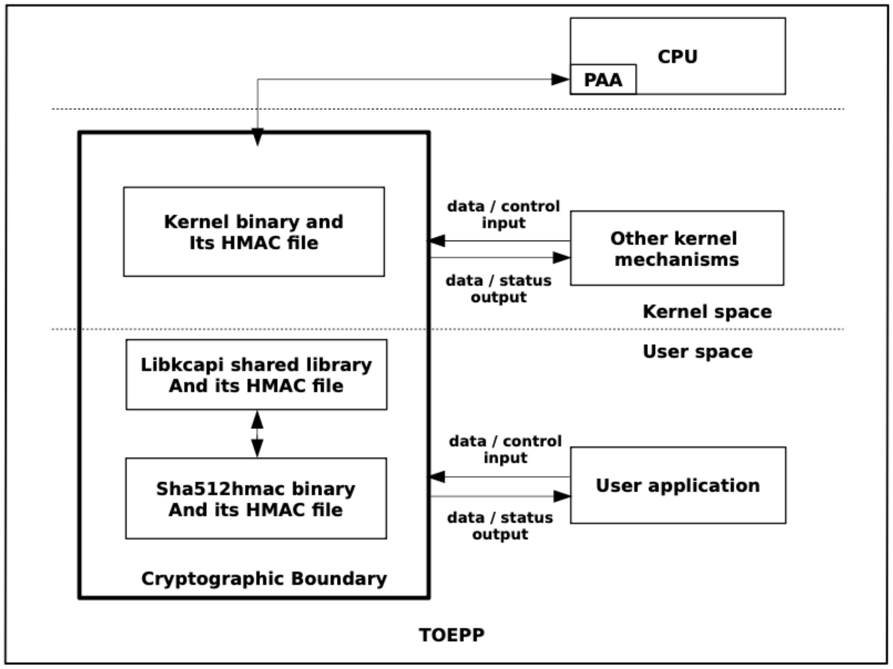

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
/boot/vmlinuz-6.1.129-1-ntap1-amd64; /usr/bin/kcapi-hasher; /lib/x86_64-linux-gnu/libkcapi.so.1.4.0	kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0	N/A	HMAC-SHA2-256 (libkcapi.so); HMAC-SHA2-512 (vmlinuz, kcapi-hasher)

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
StorageGRID 12	SG5812	Intel Xeon D-1735TR	Yes		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0
StorageGRID 12	SG5812	Intel Xeon D-1735TR	No		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0
StorageGRID 12	SG6160	Intel Xeon Gold 5318Y	Yes		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0
StorageGRID 12	SG6160	Intel Xeon Gold 5318Y	No		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0
StorageGRID 12	SG110	Intel Xeon Silver 4310	Yes		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0
StorageGRID 12	SG110	Intel Xeon Silver 4310	No		kernel 6.1.129-1-ntap1-amd64; libkcapi 1.4.0-1+ntap0

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

The module implements Processor Algorithm Acceleration (PAA) for the tested platforms listed above. There is no Processor Algorithm Implementation (PAI).

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

Operating System	Hardware Platform
StorageGRID 12	SGF6112
StorageGRID 12	SG1100
StorageGRID 12	SG5860

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

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

2.3 Excluded Components

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

2.4 Modes of Operation

Modes List and Description:

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

Table 5: Modes List and Description

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

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

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6242, A6245, A6248, A6251	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CBC-CS3	A6242, A6245, A6248, A6251	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CCM	A6242, A6245, A6251	Key Length - 128, 192, 256	SP 800-38C
AES-CFB128	A6242, A6245, A6251	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CMAC	A6242, A6245, A6251	Direction - Generation Key Length - 128, 192, 256	SP 800-38B
AES-CTR	A6242, A6245, A6248, A6251	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260, A6261, A6262, A6263	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A6242, A6244, A6245, A6247, A6248, A6250, A6251, A6253, A6254, A6256, A6258, A6260, A6261, A6263	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-GCM	A6243, A6246, A6249, A6252, A6255, A6259, A6262	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-GMAC	A6242, A6245, A6248, A6251, A6254, A6258, A6261	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 192, 256	SP 800-38D
AES-KW	A6242, A6245, A6251	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A6242, A6245, A6251	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A6242, A6245, A6248, A6251, A6254, A6257, A6258, A6261	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E
Counter DRBG	A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260, A6261, A6262, A6263	Prediction Resistance - No, Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A6242	Curve - P-256, P-384 Secret Generation Mode - testing candidates	FIPS 186-5
Hash DRBG	A6242, A6264, A6265, A6266, A6267	Prediction Resistance - No, Yes Mode - SHA-1, SHA2-256, SHA2-512	SP 800-90A Rev. 1
HMAC DRBG	A6242, A6264, A6265, A6266, A6267	Prediction Resistance - No, Yes	SP 800-90A Rev. 1

Algorithm	CAVP Cert	Properties	Reference
		Mode - SHA-1, SHA2-256, SHA2-512	
HMAC-SHA-1	A6242, A6264, A6265, A6266, A6267	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-224	A6242, A6264, A6265, A6266, A6267	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-256	A6242, A6264, A6265, A6266, A6267	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-384	A6242, A6264, A6265, A6266	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA2-512	A6242, A6264, A6265, A6266	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-224	A6242	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-256	A6242	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-384	A6242	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
HMAC-SHA3-512	A6242	Key Length - Key Length: 112-524288 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6242	Domain Parameter Generation Methods - P-256, P-384 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A6242	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDA OneStep SP800-56Cr2	A6242	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-2048 Increment 8	SP 800-56C Rev. 2
KDF SP800-108	A6242	KDF Mode - Counter Supported Lengths - Supported Lengths: 112-4096 Increment 8	SP 800-108 Rev. 1
RSA SigVer (FIPS186-4)	A6242	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A6242	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
Safe Primes Key Generation	A6242	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A6242, A6264, A6265, A6266, A6267	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-224	A6242, A6264, A6265, A6266, A6267	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-256	A6242, A6264, A6265, A6266, A6267	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-384	A6242, A6264, A6265, A6266	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA2-512	A6242, A6264, A6265, A6266	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 180-4
SHA3-224	A6242	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-256	A6242	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-384	A6242	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2	FIPS 202
SHA3-512	A6242	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
Asymmetric CKG	Key Type:Asymmetric	N/A	SP 800-133r2, section 4, example 1

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES-GCM with external IV	Encryption with external IV (not compliant to FIPS 140-3 IG C.H)
KBKDF in libkcapi	Key Derivation with implementation not tested by CAVP

Name	Use and Function
HKDF in libkcapi	Key Derivation with implementation not tested by CAVP
PBKDF2 in libkcapi	Password-Based Key Derivation with implementation not tested by CAVP
RSA PKCS#1 v1.5 with pre-hashed message	Signature generation / verification
RSA PKCS#1 v1.5	Key encapsulation / un-encapsulation (not compliant to SP 800-56Br2)
RSA primitive	Encryption / decryption (not compliant to SP 800-56Br2)

Table 8: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Encryption	BC-UnAuth	Encrypt a plaintext		AES-CBC: (A6242, A6245, A6248, A6251) AES-CBC-CS3: (A6242, A6245, A6248, A6251) AES-CFB128: (A6242, A6245, A6251) AES-CTR: (A6242, A6245, A6248, A6251) AES-ECB: (A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260,

Name	Type	Description	Properties	Algorithms
				A6261, A6262, A6263) AES-OFB: (A6242, A6245, A6251) AES-XTS Testing Revision 2.0: (A6242, A6245, A6248, A6251, A6254, A6257, A6258, A6261)
Decryption	BC-UnAuth	Decrypt a ciphertext		AES-CBC: (A6242, A6245, A6248, A6251) AES-CBC-CS3: (A6242, A6245, A6248, A6251) AES-CFB128: (A6242, A6245, A6251) AES-CTR: (A6242, A6245, A6248, A6251) AES-ECB: (A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256,

Name	Type	Description	Properties	Algorithms
				A6258, A6259, A6260, A6261, A6262, A6263) AES-OFB: (A6242, A6245, A6251) AES-XTS Testing Revision 2.0: (A6242, A6245, A6248, A6251, A6254, A6257, A6258, A6261)
Authenticated encryption	BC-Auth	Encrypt and authenticate a plaintext		AES-CCM: (A6242, A6245, A6251) AES-GCM: (A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260, A6261, A6262, A6263) AES-KW: (A6242, A6245, A6251)

Name	Type	Description	Properties	Algorithms
Authenticated decryption	BC-Auth	Decrypt and authenticate a ciphertext		AES-CCM: (A6242, A6245, A6251) AES-GCM: (A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260, A6261, A6262, A6263) AES-KW: (A6242, A6245, A6251)
Message digest	SHA	Compute a message digest		SHA-1: (A6242, A6264, A6265, A6266, A6267) SHA2-224: (A6242, A6264, A6265, A6266, A6267) SHA2-256: (A6242, A6264, A6265, A6266, A6267) SHA2-384: (A6242, A6264, A6265, A6266)

Name	Type	Description	Properties	Algorithms
				SHA2-512: (A6242, A6264, A6265, A6266) SHA3-224: (A6242) SHA3-256: (A6242) SHA3-384: (A6242) SHA3-512: (A6242)
Message authentication code generation	MAC	Compute a MAC tag		AES-CMAC: (A6242, A6245, A6251) AES-GMAC: (A6242, A6245, A6248, A6251, A6254, A6258, A6261) HMAC-SHA-1: (A6242, A6264, A6265, A6266, A6267) HMAC-SHA2-224: (A6242, A6264, A6265, A6266, A6267) HMAC-SHA2-256: (A6242, A6264, A6265, A6266, A6267) HMAC-SHA2-384: (A6242, A6264, A6265, A6266) HMAC-SHA2-512: (A6242, A6264, A6265, A6266)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA3-224: (A6242) HMAC-SHA3-256: (A6242) HMAC-SHA3-384: (A6242) HMAC-SHA3-512: (A6242)
Message authentication code verification	MAC	Verify a MAC tag		AES-GMAC: (A6242, A6245, A6248, A6251, A6254, A6258, A6261)
Key-based key derivation	KBKDF	Derive keying material from a key-derivation key		KDF SP800-108: (A6242)
Key-establishment key derivation	KAS-56CKDF	Derive keying material from a shared secret		KDA OneStep SP800-56Cr2: (A6242)
Random number generation	DRBG	Generate random bytes		Counter DRBG: (A6242, A6243, A6244, A6245, A6246, A6247, A6248, A6249, A6250, A6251, A6252, A6253, A6254, A6255, A6256, A6258, A6259, A6260, A6261, A6262, A6263) Hash DRBG: (A6242, A6264, A6265, A6266,

Name	Type	Description	Properties	Algorithms
				A6267) HMAC DRBG: (A6242, A6264, A6265, A6266, A6267)
Shared secret computation	KAS-SSC	Compute a shared secret	FFC security strength:112-200 bits ECC security strength:128, 192 bits FFC scheme:dhEphem ECC scheme:ephemeralUnified KAS role:initiator, responder Compliance:FIPS 140-3 IG D.F Scenario 2(1)	KAS-FFC-SSC Sp800-56Ar3: (A6242) KAS-ECC-SSC Sp800-56Ar3: (A6242)
Digital signature verification	DigSig-SigVer	Verify a digital signature on a message		RSA SigVer (FIPS186-4): (A6242) RSA SigVer (FIPS186-5): (A6242)
Key pair generation	AsymKeyPair-KeyGen CKG	Generate an asymmetric key pair		Safe Primes Key Generation: (A6242) ECDSA KeyGen (FIPS186-5): (A6242) Asymmetric CKG: () Key Type: Asymmetric

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES-GCM IV

The module implements AES-GCM IV generation in the context of IPsec, TLS 1.2, and TLS 1.3, compliant with RFC 4106, RFC 5288, and RFC 8446 respectively. These methods fall under Scenario 1 (“TLS/DTLS 1.2 protocol IV generation” and “IPsec-v3 protocol IV generation”) and Scenario 5 (“Provisions of an industry protocol supporting AES-GCM encryption, not included among the acceptable protocols in scenario 1”) of FIPS 140-3 IG C.H. The module is also compliant with SP 800-52r2 Section 3.3.1. IVs generated using these mechanisms may only be used in the context of AES-GCM encryption within their respective protocols.

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 does not implement TLS. 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 5288 (TLS 1.2) or RFC 8446 (TLS 1.3) to establish the cryptographic keys and initial values of the initialization vectors. For both TLS 1.2 and TLS 1.3, the nonce_explicit part of the IV does not exhaust the maximum number of possible values for a given session key. The design of the TLS protocol implicitly ensures that the nonce_explicit, or counter portion of the IV does not exhaust the maximum number of possible values for a given encryption key. In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES-GCM key encryption or decryption under this scenario shall be established.

Finally, 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 this document.

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 RSA

All supported modulus sizes for RSA signature verification have been CAVP tested.

2.7.4 SP 800-56Ar3 Assurances

The module does not establish SSPs using an approved key agreement scheme (KAS). However, it does offer some or all of the underlying KAS cryptographic functionality to be used by an external operator/application as part of an approved KAS.

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

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

2.7.5 Legacy Use

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

These legacy algorithms can only be used on data that was generated prior to the Legacy Date specified in FIPS 140-3 IG C.M.

2.8 RBG and Entropy

Cert Number	Vendor Name
E223	NetApp, Inc.

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
NetApp StorageGRID Kernel CPU Time Jitter RNG Entropy Source	Non-Physical	See Table 3	256 bits	256 bits	SHA3-256 (A6242)

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 E223 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 asymmetric key pair generation 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). 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 5.6.1.1.3 of SP 800-56Ar3 (“Extra Random Bits”) is used.
- EC key pair generation: compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-5. 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 in Section 2.6**.**

2.11 Industry Protocols

AES-GCM with internal IV generation in the approved mode is compliant with RFC 4106, RFC 5288, and RFC 8446 and shall only be used in conjunction with the IPsec, TLS 1.2, or TLS 1.3, protocols.

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

No other parts of the NVMe, Bluetooth, IPsec, or TLS 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 input sockets, SOL_TLS type input sockets
N/A	Data Output	API data output parameters, AF_ALG type output sockets, SOL_TLS type output sockets, /proc/sys/crypto virtual files
N/A	Control Input	API function calls, API control input parameters, AF_ALG type input sockets, SOL_TLS type input sockets
N/A	Status Output	API return values, AF_ALG type output sockets, SOL_TLS type output 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, AF_ALG type socket message types, and SOL_TLS socket types. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement any authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	None

Table 13: Roles

No support is provided for multiple concurrent operators.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Encryption	Encrypt a plaintext	crypto_skcipher_setkey returns 0	AES key, plaintext, IV (if required)	Ciphertext	Encryption	Crypto Officer - AES key: W,E
Decryption	Decrypt a ciphertext	crypto_skcipher_setkey returns 0	AES key, ciphertext, IV (if required)	Plaintext	Decryption	Crypto Officer - AES key: W,E
Authenticated encryption	Encrypt a plaintext in an authenticated mode	AES-GCM: crypto_aead_get_flags(tfm) has CRYPTO_ALG_FIPS140_COMPLIANT set; Others: crypto_aead_setkey returns 0	AES key, plaintext, IV (CCM/GCM)	Ciphertext, MAC tag (CCM/GCM)	Authenticated encryption	Crypto Officer - AES key: W,E
Authenticated decryption	Decrypt a ciphertext in an authenticated mode	crypto_aead_setkey returns 0	AES key, ciphertext, IV (CCM/GCM), MAC tag (CCM/GCM)	Plaintext or failure	Authenticated decryption	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message digest	Compute a message digest	crypto_shash_init returns 0	Message	Digest value	Message digest	Crypto Officer
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	Crypto Officer - AES key: W,E - HMAC key: W,E
Message authentication code verification	Verify a MAC tag	crypto_shash_init returns 0	AES key, message, MAC tag	Pass/fail	Message authentication code verification	Crypto Officer - AES key: W,E
Key-based key derivation	Derive keying material from a key-derivation key	crypto_kdf108_ctr_generate returns 0	Key-derivation key, output length	Derived key	Key-based key derivation	Crypto Officer - Key-derivation key: W,E - Derived key: G,R
Key-establishment key derivation	Derive keying material from a shared secret	crypto_kdf108_ctr_generate returns 0	Shared secret, output length	Derived key	Key-establishment key derivation	Crypto Officer - Shared secret: W,E - Derived key: G,R
Random number generation	Generate random bytes	crypto_rng_get_bytes returns 0	Output length	Random bytes	Random number generation	Crypto Officer - Entropy input: G,E,Z - HMAC_DRBG Seed: G,E,Z - HMAC_DRBG internal state (V, Key): G,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Hash_DRBG Seed: G,E,Z - Hash_DRBG internal state (V, C): G,W,E - CTR_DRBG Seed: G,E,Z - CTR_DRBG internal state (V, Key): G,W,E
Shared secret computation	Compute a shared secret	crypto_kpp_compute_shared_secret returns 0	Owner private key, peer public key	Shared secret	Shared secret computation	Crypto Officer - DH private key: W,E - DH public key: W,E - EC private key: W,E - EC public 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	Group or curve	Key pair	Key pair generation	Crypto Officer - DH private key: G,R - DH public key: G,R - EC private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key: G,R - EC public key: G,R - Intermediate key generation value: G,E,Z
Kernel TLS encryption	Perform TLS bulk data encryption	AES-GCM: crypto_aead_get_flags(aead) has CRYPTO_ALG_FIPS140_COMPLIANT set; Others: setsockopt for SOL_TLS returns 0	AES key, plaintext	Encrypted TLS record	Authenticated encryption	Crypto Officer - AES key: W,E
Kernel TLS decryption	Perform TLS bulk data decryption	setsockopt for SOL_TLS returns 0	AES key, encrypted TLS record	Plaintext or failure	Authenticated decryption	Crypto Officer - AES key: W,E
Error detection code	Compute an EDC (crc32, crc32c, crct10dif)	None	Message	EDC	None	Crypto Officer
Compression	Compress data (deflate, lz4, lz4hc, lzo, zlib-deflate, zstd)	None	Data	Compressed data	None	Crypto Officer
Generic system call	Use the kernel to perform various non-cryptographic operations	None	Identifier, various arguments	Various return values	None	Crypto Officer
Show version	Return the module	None	N/A	Module name	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	name and version information			and version		
Show status	Return the module status	None	N/A	Module status	None	Crypto Officer
Self-test	Perform the CASTs and integrity tests	None	N/A	Pass/Fail	Encryption Decryption Authenticated encryption Authenticated decryption Message digest Message authentication code verification Message authentication code generation Key-based key derivation Key-establishment key derivation Random number generation Shared secret	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					computation Digital signature verification Key pair generation	
Zeroization	Zeroize all SSPs	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - Key-derivation key: Z - Shared secret: Z - Derived key: Z - Entropy input: Z - HMAC_DRBG Seed: Z - HMAC_DRBG internal state (V, Key): Z - Hash_DRBG Seed: Z - Hash_DRBG internal state (V, C): Z - CTR_DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Seed: Z - CTR_DRBG internal state (V, Key): Z - DH public key: Z - DH private key: Z - EC public key: Z - EC private key: Z - Intermediate key generation value: 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	Encrypt and authenticate a plaintext using AES-GCM with an external IV	AES-GCM with external IV	Crypto Officer
Key derivation (libkcapi)	Derive a key from a key-derivation key, shared secret, or password	KBKDF in libkcapi HKDF in libkcapi PBKDF2 in libkcapi	Crypto Officer
Pre-hashed message signature generation	Generate a digital signature for a pre-hashed message	RSA PKCS#1 v1.5 with pre-hashed message	Crypto Officer

Name	Description	Algorithms	Role
Pre-hashed message signature verification	Verify a digital signature for a pre-hashed message	RSA PKCS#1 v1.5 with pre-hashed message	Crypto Officer
Key encapsulation	Key encapsulation using RSA PKCS#1 v1.5	RSA PKCS#1 v1.5	Crypto Officer
Key un-encapsulation	Key un-encapsulation using RSA PKCS#1 v1.5	RSA PKCS#1 v1.5	Crypto Officer
Encryption primitive	Compute the RSA encryption primitive	RSA primitive	Crypto Officer
Decryption primitive	Compute the RSA decryption primitive	RSA primitive	Crypto Officer

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

On system boot, the sha512hmac binary first performs an integrity test on itself and the libkcapi library, using the HMAC-SHA2-512 and HMAC-SHA2-256 algorithms (respectively) implemented by the module. Then, the sha512hmac binary performs an integrity test on the kernel binary using the HMAC-SHA2-512 algorithm. These tests are all performed using a key hardcoded in the sha512hmac binary, by recomputing the MAC tags and verifying they are equal to the MAC tags specified in the .hmac file.

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 (i.e. rebooting the system), which will perform (among others) the software integrity tests.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied: The operating system provides process isolation and memory protection mechanisms that ensure appropriate separation for memory access among the processes on the system. Each process has control over its own data and uncontrolled access to the data of other processes is prevented.

6.2 Configuration Settings and Restrictions

The module shall be installed as stated in Section 11.

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

7 Physical Security

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

8 Non-Invasive Security

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

9 Sensitive Security Parameters Management

9.1 Storage Areas

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

Table 16: Storage Areas

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

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API input parameters	Operator calling application TOEPP	RAM	Plaintext	Manual	Electronic	
AF_ALG type input sockets	Operator calling application TOEPP	RAM	Plaintext	Manual	Electronic	
SOL_TLS type input sockets	Operator calling application TOEPP	RAM	Plaintext	Manual	Electronic	
API output parameters	RAM	Operator calling application TOEPP	Plaintext	Manual	Electronic	
AF_ALG type output sockets	RAM	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

Zeroization Method	Description	Rationale	Operator Initiation
		The completion of the zeroization routine indicates that the zeroization procedure succeeded.	crypto_free_ahash; Key-derivation key: crypto_free_shash; Shared secret: crypto_free_shash; Entropy input: crypto_free_rng; DRBG seed: crypto_free_rng; DRBG internal state: crypto_free_rng; DH public key & DH private key: crypto_free_kpp; EC public key & EC private key: crypto_free_kpp; RSA public key: public_key_free
Remove power from the module	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed. Module power off indicates that the zeroization procedure succeeded.	By removing power
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A

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	Symmetric key used for AES operations	128, 256 bits (AES-XTS); 128, 192, 256 bits (others) - 128, 256 bits (AES-XTS); 128, 192, 256	Symmetric key - CSP			Encryption Decryption Authenticated encryption Authenticated decryption Message authentication code verification Message authenticati

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		bits (others)				on code generation
HMAC key	Symmetric key used for HMAC operations	112-524288 bits - 112-256 bits	Authentication key - CSP			Message authentication code verification Message authentication code generation
Key-derivation key	Symmetric key used in performing key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key-based key derivation
Shared secret	Shared secret generated by (EC) Diffie-Hellman	P-256, P-384 / ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 128-8192 bits	Shared secret - CSP		Shared secret computation	Key-establishment key derivation
Derived key	Symmetric key produced by a key derivation service.	112-4096 - 112-256	Symmetric key - CSP	Key-based key derivation Key-establishment key derivation		
Entropy input	Entropy input used to seed DRBGs	128-384 bits - 128-384 bits	Entropy input - CSP	Random number generation		Random number generation
HMAC_DRBG Seed	DRBG seed derived from entropy input and additional data	440, 888 bits - 128, 256 bits	Seed - CSP	Random number generation		Random number generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
CTR_DRBG Seed	DRBG seed derived from Entropy Input and additional data	256, 320, 384 bits - 128, 192, 256 bits	Seed - CSP	Random number generation		Random number generation
Hash_DRBG Seed	DRBG seed derived from Entropy Input and additional data	440, 888 bits - 128, 256 bits	Seed - CSP	Random number generation		Random number generation
HMAC_DRBG internal state (V, Key)	Internal state of HMAC_DRBG	320, 512, 1024 bits - 128, 256 bits	Internal state - CSP	Random number generation		Random number generation
CTR_DRBG internal state (V, Key)	Internal state of CTR_DRBG	256, 320, 348 bits - 128, 192, 256 bits	Internal state - CSP	Random number generation		Random number generation
Hash_DRBG internal state (V, C)	Internal state of Hash_DRBG	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random number generation		Random number generation
DH private key	Private key used for Diffie-Hellman	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - 112-200 bits	Private key - CSP	Key pair generation		Shared secret computation
DH public key	Public key used for Diffie-Hellman	ffdhe2048, ffdhe3072, ffdhe4096,	Public key - PSP	Key pair generation		Shared secret computation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		ffdhe6144, ffdhe8192 - 112-200 bits				
EC private key	Private key used for EC Diffie-Hellman	P-256, P-384 - 128, 192 bits	Private key - CSP	Key pair generation		Shared secret computation
EC public key	Public key used for EC Diffie-Hellman	P-256, P-384 - 128, 192 bits	Public key - PSP	Key pair generation		Shared secret computation
Intermediate key generation value	Temporary value generated during key pair generation services	256-8192 bits - 112-200 bits	Intermediate value - CSP	Key pair generation		

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters AF_ALG type input sockets SOL_TLS type input sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	
HMAC key	API input parameters AF_ALG type input sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	
Key-derivation key	API input parameters AF_ALG type input sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	
Shared secret	API input parameters API output parameters AF_ALG type input	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	sockets AF_ALG type output sockets				
Derived key	API output parameters AF_ALG type output sockets	RAM:Plaintext	For the duration of the service	Automatic	Key-derivation key:Derived From Shared secret:Derived From
Entropy input		RAM:Plaintext	From generation until DRBG seed/reseed	Automatic	DRBG seed:Derivation Of
HMAC_DRBG Seed		RAM:Plaintext	From generation until HMAC_DRBG Seed is created	Automatic	Entropy input:Derived From DRBG internal state (V, Key):Derivation Of DRBG internal state (V, C):Derivation Of
CTR_DRBG Seed		RAM:Plaintext	From generation until DRBG seed is created	Automatic	Entropy input:Derived From
Hash_DRBG Seed		RAM:Plaintext	From generation until DRBG seed is created	Automatic	Entropy input:Derived From
HMAC_DRBG internal state (V, Key)		RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	HMAC_DRBG Seed:Derived From
CTR_DRBG internal state (V, Key)		RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	CTR_DRBG Seed:Derived From
Hash_DRBG internal state (V, C)		RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	Hash_DRBG Seed:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DH private key	API input parameters API output parameters AF_ALG type input sockets AF_ALG type output sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	DH public key:Paired With
DH public key	API input parameters API output parameters AF_ALG type input sockets AF_ALG type output sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	DH private key:Paired With
EC private key	API input parameters API output parameters AF_ALG type input sockets AF_ALG type output sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	EC public key:Paired With
EC public key	API input parameters API output parameters AF_ALG type input sockets AF_ALG type output sockets	RAM:Plaintext	Until cipher handle is freed or module is powered off	Free cipher handle Remove power from the module	EC private key:Paired With
Intermediate key generation value		RAM:Plaintext	For the duration of the service	Automatic	

Table 20: SSP Table 2

9.5 Transitions

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

10 Self-Tests

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 does not return control to the calling application until the tests are completed. If any of the self-tests fails, the module immediately transitions to the error state.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-512 - kcapi-hasher	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational	Integrity test for kcapi-hasher binary
HMAC-SHA2-256 - libkcapi	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational	Integrity test for libkcapi shared library
HMAC-SHA2-512 - vmlinuz	128-bit key	Message Authentication	SW/FW Integrity	Module becomes operational	Integrity test for vmlinuz binary

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.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CBC Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CBC (AESNI_ASM) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CBC (AESNI_ASM) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CBC (AESNI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CBC (AESNI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC-CS3 Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CBC-CS3 Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CBC-CS3 (AESNI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CBC-CS3 (AESNI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CCM Authenticated encryption	128, 192, 256-bit keys; 56, 64, 72, 80, 88, 96, 112, 128-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-CCM Authenticated decryption	128, 192, 256-bit keys; 56, 64, 72, 80, 88, 96, 112, 128-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-CCM (AESNI_C) Authenticated encryption	128, 192, 256-bit keys; 56, 64, 72, 80, 88, 96, 112, 128-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-CCM (AESNI_C) Authenticated decryption	128, 192, 256-bit keys; 56, 64, 72, 80, 88, 96, 112, 128-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-CFB128 Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CFB128 Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CFB128 (AESNI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CFB128 (AESNI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CTR Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CTR Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-CTR (AESNI_ASM) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-CTR (AESNI_ASM) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-ECB Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-ECB Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-ECB (CTI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-ECB (CTI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-ECB (AESNI_ASM) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-ECB (AESNI_ASM) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-ECB (AESNI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-ECB (AESNI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-GCM Authenticated encryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-GCM Authenticated decryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (AESNI_ASM) Authenticated encryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-GCM (AESNI_ASM) Authenticated decryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-GCM (AESNI_AVX) Authenticated encryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-GCM (AESNI_AVX) Authenticated decryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-GCM (VAES_AVX10_256) Authenticated encryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-GCM (VAES_AVX10_256) Authenticated decryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-GCM (VAES_AVX10_512) Authenticated encryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated encryption	Module initialization
AES-GCM (VAES_AVX10_512) Authenticated decryption	128, 192, 256-bit keys; 96-bit IVs	KAT	CAST	Module is operational	Authenticated decryption	Module initialization
AES-OFB Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-OFB Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-OFB (AESNI_C) Encryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-OFB (AESNI_C) Decryption	128, 192, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-XTS Testing Revision 2.0 Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-XTS Testing Revision 2.0 (AESNI_ASM) Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 (AESNI_ASM) Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-XTS Testing Revision 2.0 (AESNI_AVX) Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 (AESNI_AVX) Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX2) Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX2) Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX10_256) Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX10_256) Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX10_512) Encryption	128, 256-bit keys	KAT	CAST	Module is operational	Encryption	Module initialization
AES-XTS Testing Revision 2.0 (VAES_AVX10_512) Decryption	128, 256-bit keys	KAT	CAST	Module is operational	Decryption	Module initialization
SHA-1	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (SSSE3)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA-1 (AVX)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA-1 (AVX2)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA-1 (SHA_NI)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-224	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-224 (SSSE3)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-224 (AVX)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-224 (AVX2)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-224 (SHA_NI)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-256	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-256 (SSSE3)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-256 (AVX)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-256 (AVX2)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-256 (SHA_NI)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-384	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-384 (SSSE3)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-384 (AVX)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-384 (AVX2)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-512	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-512 (SSSE3)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-512 (AVX)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA2-512 (AVX2)	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA3-224	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA3-256	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA3-384	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
SHA3-512	0-65536-bit messages	KAT	CAST	Module is operational	Message Digest	Module initialization
AES-CMAC	128, 192, 256-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
AES-CMAC (AESNI_C)	128, 192, 256-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA-1	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA-1 (SHA_NI)	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA2-224	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA2-224 (SHA_NI)	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Before integrity test
HMAC-SHA2-256 (SHA_NI)	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Before integrity test
HMAC-SHA2-384	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA2-384 (AVX2)	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA2-512	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA2-512 (AVX2)	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA3-224	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA3-256	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA3-384	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
HMAC-SHA3-512	112-524288-bit keys	KAT	CAST	Module is operational	Message Authentication	Module initialization
KDF SP800-108	SHA2-256	KAT	CAST	Module is operational	Key derivation	Module initialization
KDA OneStep SP800-56Cr2	SHA2-256	KAT	CAST	Module is operational	Key derivation	Module initialization
Counter DRBG	AES-128, AES-192, and AES-256 with/without prediction resistance	KAT	CAST	Module is operational	Instantiate, seed, reseed, generate (compliant to SP 800-90Ar1 Section 11.3)	Module initialization
Hash DRBG	SHA-1, SHA-256, and SHA-512	KAT	CAST	Module is operational	Instantiate, seed, reseed, generate	Module initialization

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	with/without prediction resistance				(compliant to SP 800-90Ar1 Section 11.3)	
HMAC DRBG	SHA-1, SHA-256, and SHA-512 with/without prediction resistance	KAT	CAST	Module is operational	Instantiate, seed, reseed, generate (compliant to SP 800-90Ar1 Section 11.3)	Module initialization
KAS-FFC-SSC Sp800-56Ar3	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	KAT	CAST	Module is operational	Shared secret computation	Module initialization
KAS-ECC-SSC Sp800-56Ar3	P-256, P-384	KAT	CAST	Module is operational	Shared secret computation	Module initialization
RSA SigVer (FIPS186-5)	PKCS#1 v1.5 with SHA-224, SHA-256, SHA-384, SHA-512 and 2048, 3072, 4096-bit keys	KAT	CAST	Module is operational	Signature verification	Module initialization
Entropy Source Start Up APT	Cutoff C = 325; Window size = 512	APT	CAST	Entropy source is operational	Entropy source start-up test on 1024 samples	Entropy source initialization
Entropy Source Start Up RCT	Cutoff C = 31	RCT	CAST	Entropy source is operational	Entropy source start-up test on 1024 samples	Entropy source initialization
Entropy Source Continuous APT	Permanent cutoff C	APT	CAST	jent_kcapi_random returns 0	Entropy source	Continuously as

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	= 355; Window size = 512				continuous test	entropy is requested
Entropy Source Continuous RCT	Permanent cutoff C = 61	RCT	CAST	jent_kcapi_random returns 0	Entropy source continuous test	Continuously as entropy is requested
Safe Primes Key Generation	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	PCT	PCT	Key pair generation is successful	SP 800-56Ar3 Section 5.6.2.1.4	Key pair generation
ECDSA KeyGen (FIPS186-5)	P-256, P-384	PCT	PCT	Key pair generation is successful	SP 800-56Ar3 Section 5.6.2.1.4	Key pair generation

Table 22: Conditional Self-Tests

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

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 - kcapi-hasher	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 - libkcapi	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-512 - vmlinuz	Message Authentication	SW/FW Integrity	On demand	Manually

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Encryption	KAT	CAST	On demand	Manually
AES-CBC Decryption	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (AESNI_ASM) Encryption	KAT	CAST	On demand	Manually
AES-CBC (AESNI_ASM) Decryption	KAT	CAST	On demand	Manually
AES-CBC (AESNI_C) Encryption	KAT	CAST	On demand	Manually
AES-CBC (AESNI_C) Decryption	KAT	CAST	On demand	Manually
AES-CBC-CS3 Encryption	KAT	CAST	On demand	Manually
AES-CBC-CS3 Decryption	KAT	CAST	On demand	Manually
AES-CBC-CS3 (AESNI_C) Encryption	KAT	CAST	On demand	Manually
AES-CBC-CS3 (AESNI_C) Decryption	KAT	CAST	On demand	Manually
AES-CCM Authenticated encryption	KAT	CAST	On demand	Manually
AES-CCM Authenticated decryption	KAT	CAST	On demand	Manually
AES-CCM (AESNI_C) Authenticated encryption	KAT	CAST	On demand	Manually
AES-CCM (AESNI_C) Authenticated decryption	KAT	CAST	On demand	Manually
AES-CFB128 Encryption	KAT	CAST	On demand	Manually
AES-CFB128 Decryption	KAT	CAST	On demand	Manually
AES-CFB128 (AESNI_C) Encryption	KAT	CAST	On demand	Manually
AES-CFB128 (AESNI_C) Decryption	KAT	CAST	On demand	Manually
AES-CTR Encryption	KAT	CAST	On demand	Manually
AES-CTR Decryption	KAT	CAST	On demand	Manually
AES-CTR (AESNI_ASM) Encryption	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CTR (AESNI_ASM) Decryption	KAT	CAST	On demand	Manually
AES-ECB Encryption	KAT	CAST	On demand	Manually
AES-ECB Decryption	KAT	CAST	On demand	Manually
AES-ECB (CTI_C) Encryption	KAT	CAST	On demand	Manually
AES-ECB (CTI_C) Decryption	KAT	CAST	On demand	Manually
AES-ECB (AESNI_ASM) Encryption	KAT	CAST	On demand	Manually
AES-ECB (AESNI_ASM) Decryption	KAT	CAST	On demand	Manually
AES-ECB (AESNI_C) Encryption	KAT	CAST	On demand	Manually
AES-ECB (AESNI_C) Decryption	KAT	CAST	On demand	Manually
AES-GCM Authenticated encryption	KAT	CAST	On demand	Manually
AES-GCM Authenticated decryption	KAT	CAST	On demand	Manually
AES-GCM (AESNI_ASM) Authenticated encryption	KAT	CAST	On demand	Manually
AES-GCM (AESNI_ASM) Authenticated decryption	KAT	CAST	On demand	Manually
AES-GCM (AESNI_AVX) Authenticated encryption	KAT	CAST	On demand	Manually
AES-GCM (AESNI_AVX) Authenticated decryption	KAT	CAST	On demand	Manually
AES-GCM (VAES_AVX10_256) Authenticated encryption	KAT	CAST	On demand	Manually
AES-GCM (VAES_AVX10_256) Authenticated decryption	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (VAES_AVX10_512) Authenticated encryption	KAT	CAST	On demand	Manually
AES-GCM (VAES_AVX10_512) Authenticated decryption	KAT	CAST	On demand	Manually
AES-OFB Encryption	KAT	CAST	On demand	Manually
AES-OFB Decryption	KAT	CAST	On demand	Manually
AES-OFB (AESNI_C) Encryption	KAT	CAST	On demand	Manually
AES-OFB (AESNI_C) Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 Encryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (AESNI_ASM) Encryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (AESNI_ASM) Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (AESNI_AVX) Encryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (AESNI_AVX) Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (VAES_AVX2) Encryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (VAES_AVX2) Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (VAES_AVX10_256) Encryption	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-XTS Testing Revision 2.0 (VAES_AVX10_256) Decryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (VAES_AVX10_512) Encryption	KAT	CAST	On demand	Manually
AES-XTS Testing Revision 2.0 (VAES_AVX10_512) Decryption	KAT	CAST	On demand	Manually
SHA-1	KAT	CAST	On demand	Manually
SHA-1 (SSSE3)	KAT	CAST	On demand	Manually
SHA-1 (AVX)	KAT	CAST	On demand	Manually
SHA-1 (AVX2)	KAT	CAST	On demand	Manually
SHA-1 (SHA_NI)	KAT	CAST	On demand	Manually
SHA2-224	KAT	CAST	On demand	Manually
SHA2-224 (SSSE3)	KAT	CAST	On demand	Manually
SHA2-224 (AVX)	KAT	CAST	On demand	Manually
SHA2-224 (AVX2)	KAT	CAST	On demand	Manually
SHA2-224 (SHA_NI)	KAT	CAST	On demand	Manually
SHA2-256	KAT	CAST	On demand	Manually
SHA2-256 (SSSE3)	KAT	CAST	On demand	Manually
SHA2-256 (AVX)	KAT	CAST	On demand	Manually
SHA2-256 (AVX2)	KAT	CAST	On demand	Manually
SHA2-256 (SHA_NI)	KAT	CAST	On demand	Manually
SHA2-384	KAT	CAST	On demand	Manually
SHA2-384 (SSSE3)	KAT	CAST	On demand	Manually
SHA2-384 (AVX)	KAT	CAST	On demand	Manually
SHA2-384 (AVX2)	KAT	CAST	On demand	Manually
SHA2-512	KAT	CAST	On demand	Manually
SHA2-512 (SSSE3)	KAT	CAST	On demand	Manually
SHA2-512 (AVX)	KAT	CAST	On demand	Manually
SHA2-512 (AVX2)	KAT	CAST	On demand	Manually
SHA3-224	KAT	CAST	On demand	Manually
SHA3-256	KAT	CAST	On demand	Manually
SHA3-384	KAT	CAST	On demand	Manually
SHA3-512	KAT	CAST	On demand	Manually
AES-CMAC	KAT	CAST	On demand	Manually
AES-CMAC (AESNI_C)	KAT	CAST	On demand	Manually
HMAC-SHA-1	KAT	CAST	On demand	Manually
HMAC-SHA-1 (SHA_NI)	KAT	CAST	On demand	Manually
HMAC-SHA2-224	KAT	CAST	On demand	Manually
HMAC-SHA2-224 (SHA_NI)	KAT	CAST	On demand	Manually
HMAC-SHA2-256	KAT	CAST	On demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (SHA_NI)	KAT	CAST	On demand	Manually
HMAC-SHA2-384	KAT	CAST	On demand	Manually
HMAC-SHA2-384 (AVX2)	KAT	CAST	On demand	Manually
HMAC-SHA2-512	KAT	CAST	On demand	Manually
HMAC-SHA2-512 (AVX2)	KAT	CAST	On demand	Manually
HMAC-SHA3-224	KAT	CAST	On demand	Manually
HMAC-SHA3-256	KAT	CAST	On demand	Manually
HMAC-SHA3-384	KAT	CAST	On demand	Manually
HMAC-SHA3-512	KAT	CAST	On demand	Manually
KDF SP800-108	KAT	CAST	On demand	Manually
KDA OneStep SP800-56Cr2	KAT	CAST	On demand	Manually
Counter DRBG	KAT	CAST	On demand	Manually
Hash DRBG	KAT	CAST	On demand	Manually
HMAC DRBG	KAT	CAST	On demand	Manually
KAS-FFC-SSC Sp800-56Ar3	KAT	CAST	On demand	Manually
KAS-ECC-SSC Sp800-56Ar3	KAT	CAST	On demand	Manually
RSA SigVer (FIPS186-5)	KAT	CAST	On demand	Manually
Entropy Source Start Up APT	APT	CAST	On demand	Manually
Entropy Source Start Up RCT	RCT	CAST	On demand	Manually
Entropy Source Continuous APT	APT	CAST	On demand	Manually
Entropy Source Continuous RCT	RCT	CAST	On demand	Manually
Safe Primes Key Generation	PCT	PCT	On demand	Manually
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	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, CASTs and entropy source start-up tests can be invoked on demand by unloading and subsequently re-initializing the module. The PCTs can be invoked on demand by requesting the key pair generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the StorageGRID 12 operating system. The StorageGRID Grid Manager is used to install the module:

1. Open the sidebar menu and click CONFIGURATION
2. Under Security, click Security settings
3. Install the FIPS module using one of the following options:
    a. Use the “FIPS strict” policy
    b. Configure and use a custom policy with the “fipsMode” key set to “true”
4. After enabling the policy, a rolling reboot must be performed; the module is not considered installed until a reboot is performed

11.2 Administrator Guidance

After installation of module, the Crypto Officer must use the StorageGRID Grid Manager to verify the correct name and version of the module:

1. Open the sidebar menu and click SUPPORT
2. Under Tools, click Diagnostics
3. Find the “FIPS module versions” diagnostic and verify the FIPS module name and FIPS module version are listed as follows:
   NetApp StorageGRID Kernel Crypto API 6.1.129-1-ntap1-amd64
   kcapi-tools 1.4.0-1+ntap0
   libkcapi1:amd64 1.4.0-1+ntap0

The FIPS module is only installed on a given node if the aforementioned names and versions are displayed for that node.

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 Design and Rules

Not applicable.

11.5 Maintenance Requirements

Not applicable.

11.6 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory.

12 Mitigation of Other Attacks

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

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
CBC-CS3	Cipher Block Chaining with Ciphertext Stealing 3
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
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ESV	Entropy Source Validation
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
IG	Implementation Guidance
IPsec	Internet Protocol Security
IV	Initialization Vector
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
KDA	Key Derivation Algorithm
KDF	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
PBKDF	Password-Based Key Derivation Function
PCT	Pair-wise Consistency Test
PKCS	Public-Key Cryptography Standard
PSP	Public Security Parameter
PUB	Processing Standards Publication
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
TOEPP	Tested Operational Environment’s Physical Perimeter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

B References

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

FIPS 140-3 IG **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**
https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf

FIPS 180-4 **Secure Hash Standard (SHS)**
August 2015
https://doi.org/10.6028/NIST.FIPS.180-4

FIPS 186-4 **Digital Signature Standard (DSS)**
July 2013
https://doi.org/10.6028/NIST.FIPS.186-4

FIPS 186-5 **Digital Signature Standard (DSS)**
February 2023
https://doi.org/10.6028/NIST.FIPS.186-5

FIPS 197 **Advanced Encryption Standard (AES)**
November 2001; Updated May 2023
https://doi.org/10.6028/NIST.FIPS.197-upd1

FIPS 198-1 **The Keyed-Hash Message Authentication Code (HMAC)**
July 2008
https://doi.org/10.6028/NIST.FIPS.198-1

FIPS 202 **SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions**
August 2015
https://doi.org/10.6028/NIST.FIPS.202

PKCS#1 **PKCS #1: RSA Cryptography Specifications Version 2.2**
November 2016
https://doi.org/10.17487/RFC8017

RFC 4106 **The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)**
June 2005
https://doi.org/10.17487/RFC4106

RFC 5288 **The Transport Layer Security (TLS) Protocol Version 1.2**
August 2008
https://doi.org/10.17487/RFC5246

RFC 8446 **The Transport Layer Security (TLS) Protocol Version 1.3**
August 2018
https://doi.org/10.17487/RFC8446

SP 800-38A **Recommendation for Block Cipher Modes of Operation: Methods and Techniques**
December 2001
https://doi.org/10.6028/NIST.SP.800-38A

SP800-38A-Add **Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode**
October 2010
https://doi.org/10.6028/NIST.SP.800-38A-Add

SP 800-38B **Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication**
May 2005; Updated October 2016
https://doi.org/10.6028/NIST.SP.800-38B

SP 800-38C **Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**
May 2004; Updated July 2007
https://doi.org/10.6028/NIST.SP.800-38C

SP 800-38D **Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**
November 2007
https://doi.org/10.6028/NIST.SP.800-38D

SP 800-38E **Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices**
January 2010
https://doi.org/10.6028/NIST.SP.800-38E

SP 800-38F **Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping**
December 2012
https://doi.org/10.6028/NIST.SP.800-38F

SP 800-52r2 **Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations**
August 2019
https://doi.org/10.6028/NIST.SP.800-52r2

SP 800-56Ar3 **Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography**
April 2018
https://doi.org/10.6028/NIST.SP.800-56Ar3

SP 800-56Cr2 **Recommendation for Key-Derivation Methods in Key-Establishment Schemes**
August 2020
https://doi.org/10.6028/NIST.SP.800-56Cr2

SP 800-90Ar1 **Recommendation for Random Number Generation Using Deterministic Random Bit Generators**
June 2015
https://doi.org/10.6028/NIST.SP.800-90Ar1

SP 800-90B **Recommendation for the Entropy Sources Used for Random Bit Generation**
January 2018
https://doi.org/10.6028/NIST.SP.800-90B

SP 800-108r1 **Recommendation for Key Derivation Using Pseudorandom Functions**
August 2022
https://doi.org/10.6028/NIST.SP.800-108r1-upd1

SP 800-131Ar2 **Transitioning the Use of Cryptographic Algorithms and Key Lengths**
Marcy 2019
https://doi.org/10.6028/NIST.SP.800-131Ar2

SP 800-133r2 **Recommendation for Cryptographic Key Generation**
June 2020
https://doi.org/10.6028/NIST.SP.800-133r2

SP 800-140Br1 **Cryptographic Module Validation Program (CMVP) Security Policy Requirements: CMVP Validation Authority Updates to ISO/IEC 24759 and ISO/IEC 19790 Annex B**
November 2023
https://doi.org/10.6028/NIST.SP.800-140Br1