Canonical Ltd.

Canonical Ltd. Ubuntu 24.04 OpenSSL Cryptographic Module

Version 3.0.13-0ubuntu3+Fips1

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.2

Last Updated: 2025-12-16

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

Prepared for:
Canonical Ltd.
110 Southwark Street, Blue Fin Building, 5th Floor
London, SE1 0SU
www.canonical.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 3.0.13-0ubuntu3+Fips1 of the Canonical Ltd. Ubuntu 24.04 OpenSSL 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	1
	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 Canonical Ltd. Ubuntu 24.04 OpenSSL Cryptographic Module (hereafter referred to as “the module”) is defined as a software module in a multi-chip standalone embodiment. It provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module consists of one software component, the “FIPS provider” i.e., fips.so, which implements the FIPS requirements, and the cryptographic functionality provided to the operator.

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics

Cryptographic Boundary:

Components in white are only included in the diagram for informational purposes. They are not included in the cryptographic boundary (and therefore not part of the module’s validation). For example, the kernel is responsible for managing system calls issued by the module itself, as well as other applications using the module for cryptographic services.

Tested Operational Environment’s Physical Perimeter (TOEPP):

Figure 1 shows a block diagram that represents the design of the module when the module is operational and providing services to other user space applications. In this diagram, the physical perimeter of the operational environment (a general-purpose computer on which the module is installed) is indicated by a purple dashed line. The cryptographic boundary is represented by the component in the orange block, that is, the shared library implementing the FIPS provider (fips.so).

The connecting lines indicate the flow of data between the cryptographic module and its operator application, through the logical interfaces defined in Section 3.

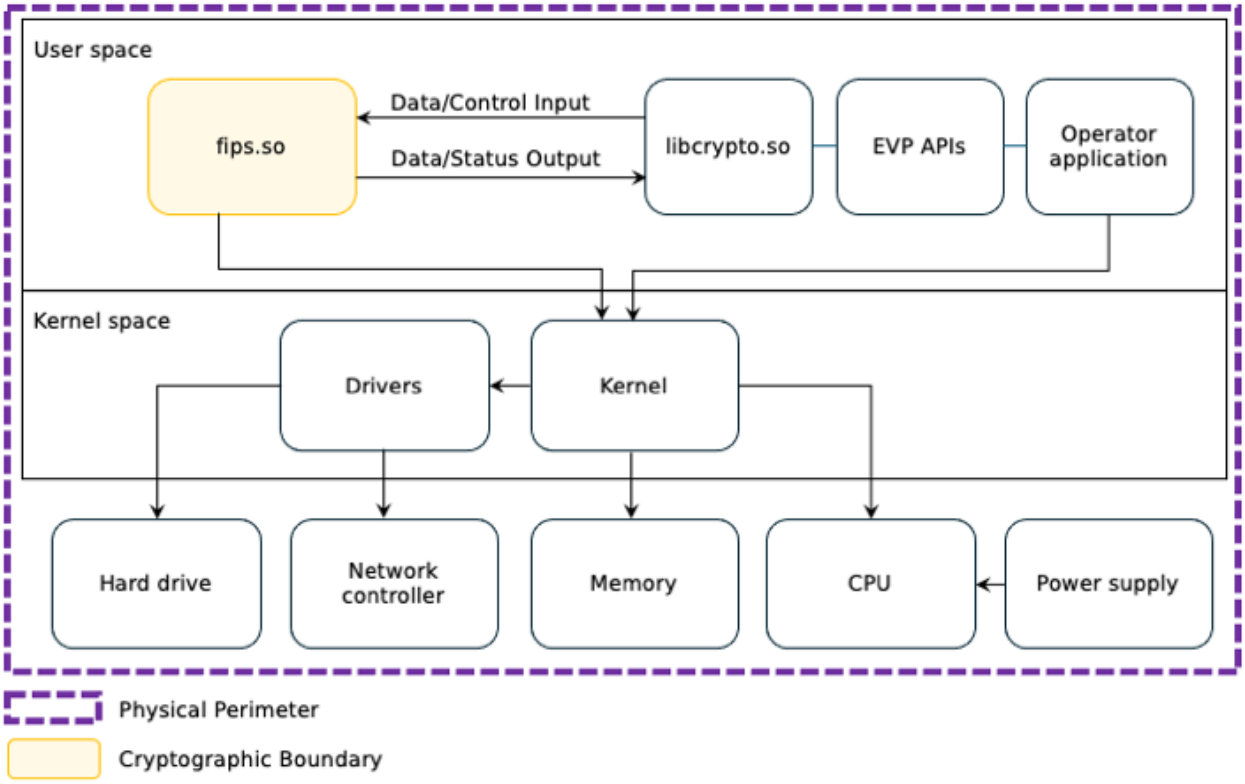

Figure 1: Block Diagram

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
fips.so on Ubuntu 24.04 with Xeon Gold 6226	3.0.13-0ubuntu3+Fips1	N/A	HMAC-SHA-256
fips.so on Ubuntu 24.04 with AWS Graviton3	3.0.13-0ubuntu3+Fips1	N/A	HMAC-SHA-256
fips.so on Ubuntu 24.04 with IBM Telum	3.0.13-0ubuntu3+Fips1	N/A	HMAC-SHA-256

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

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Ubuntu 24.04	Supermicro SYS-1019P-WTR	Intel Xeon Gold 6226	Yes	N/A	3.0.13-0ubuntu3+Fips1
Ubuntu 24.04	Supermicro SYS-1019P-WTR	Intel Xeon Gold 6226	No	N/A	3.0.13-0ubuntu3+Fips1
Ubuntu 24.04	Amazon Web Services (AWS) c7g.metal	AWS Graviton3	Yes	N/A	3.0.13-0ubuntu3+Fips1
Ubuntu 24.04	Amazon Web Services (AWS) c7g.metal	AWS Graviton3	No	N/A	3.0.13-0ubuntu3+Fips1
Ubuntu 24.04	IBM z16	IBM Telum	Yes	N/A	3.0.13-0ubuntu3+Fips1
Ubuntu 24.04	IBM z16	IBM Telum	No	N/A	3.0.13-0ubuntu3+Fips1

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

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

N/A for this module.

2.3 Excluded Components

There are no components excluded from the module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Automatically entered whenever an approved service is requested	Approved	Equivalent to the indicator of the requested service listed in section 4.3
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	Equivalent to the indicator of the requested service listed in section 4.4

Table 4: Modes List and Description

The module supports two modes of operation: (1) the approved mode of operation, in which the approved or vendor affirmed services are available as specified in the Approved Services table and (2) the non-approved mode of operation, in which the non-approved services are available as specified in the Non-Approved Services table.

Mode Change Instructions and Status:

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

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

The table below lists all implemented modes or methods of operation for the approved cryptographic algorithms of the module that are employed for approved services (Approved Services table).

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CBC-CS1	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CBC-CS2	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CBC-CS3	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CCM	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38C
AES-CFB1	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CFB128	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CFB8	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-CMAC	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38B
AES-CTR	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-ECB	A5747, A5748, A5749, A5751, A5752, A5753, A5754, A5755, A5774, A5775, A5776, A5927, A5932	-	SP 800-38A
AES-GCM	A5759, A5760, A5761, A5762, A5763, A5764, A5765, A5766, A5767, A5777, A5781, A5782, A5783, A5928, A5929, A5930	-	SP 800-38D
AES-GMAC	A5759, A5760, A5761, A5762, A5763, A5764, A5765, A5766, A5767, A5777, A5781, A5782, A5783, A5928, A5929, A5930	-	SP 800-38D
AES-KW	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38F
AES-KWP	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38F
AES-OFB	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38A
AES-XTS Testing Revision 2.0	A5747, A5748, A5749, A5774, A5775, A5776, A5927	-	SP 800-38E
Counter DRBG	A5746	-	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A5745, A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 186-5
ECDSA KeyVer (FIPS186-4)	A5745, A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 186-4
ECDSA KeyVer (FIPS186-5)	A5745, A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-4)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	FIPS 186-5
Hash DRBG	A5746	-	SP 800-90A Rev. 1
HMAC DRBG	A5746	-	SP 800-90A Rev. 1
HMAC-SHA-1	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA2-224	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA2-256	A5750, A5768, A5769, A5770, A5771, A5778, A5779, A5931	-	FIPS 198-1
HMAC-SHA2-384	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA2-512	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA2-512/224	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA2-512/256	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 198-1
HMAC-SHA3-224	A5756, A5780, A5933	-	FIPS 198-1
HMAC-SHA3-256	A5756, A5780, A5933	-	FIPS 198-1
HMAC-SHA3-384	A5756, A5780, A5933	-	FIPS 198-1
HMAC-SHA3-512	A5756, A5780, A5933	-	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A5773	-	SP 800-56A Rev. 3
KDA HKDF SP800-56Cr2	A5758	-	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A5744	-	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A5744	-	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	SP 800-135 Rev. 1
KDF SP800-108	A5772	-	SP 800-108 Rev. 1
KDF SSH (CVL)	A5751, A5752, A5753, A5754, A5755, A5932	-	SP 800-135 Rev. 1
KMAC-128	A5756, A5780, A5933	-	SP 800-185
KMAC-256	A5756, A5780, A5933	-	SP 800-185
PBKDF	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	SP 800-132

Algorithm	CAVP Cert	Properties	Reference
RSA KeyGen (FIPS186-5)	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 186-5
RSA SigGen (FIPS186-5)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	FIPS 186-5
RSA SigVer (FIPS186-4)	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 186-4
RSA SigVer (FIPS186-5)	A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933	-	FIPS 186-5
Safe Primes Key Generation	A5773	-	SP 800-56A Rev. 3
Safe Primes Key Verification	A5773	-	SP 800-56A Rev. 3
SHA-1	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA2-224	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA2-256	A5750, A5768, A5769, A5770, A5771, A5778, A5779, A5931	-	FIPS 180-4
SHA2-384	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA2-512	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA2-512/224	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA2-512/256	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	FIPS 180-4
SHA3-224	A5756, A5780, A5933	-	FIPS 202
SHA3-256	A5756, A5780, A5933	-	FIPS 202
SHA3-384	A5756, A5780, A5933	-	FIPS 202
SHA3-512	A5756, A5780, A5933	-	FIPS 202
SHAKE-128	A5756, A5780, A5933	-	FIPS 202
SHAKE-256	A5756, A5780, A5933	-	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A5750, A5768, A5769, A5770, A5771, A5778, A5931	-	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A5758	-	SP 800-135 Rev. 1

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Asymmetric Cryptographic Key Generation (CKG)	Key Type:Asymmetric	N/A	SP 800-133Rev2 section 4, example 1

Table 6: 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 (external IV)	Encryption
DSA	Signature Generation, Signature Verification, Key Pair Generation, Key Pair Verification
ECDSA with curve P-192, B-163, K-163	Key Pair Generation
ECDSA with curve B-163, K-163	Key Pair Verification
ECDSA with curve P-192	Signature Generation
ECDSA with curve B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571	Signature Generation, Signature Verification
RSA and ECDSA (pre-hashed message)	Signature Generation (pre-hashed message), Signature Verification (pre-hashed message)
RSA X9.31	Signature Generation, Signature Verification
RSA primitive	Asymmetric Encryption, Asymmetric Decryption
RSA-OAEP	Asymmetric Encryption, Asymmetric Decryption
RSASVE	Secret Value Encapsulation, Secret Value Decapsulation

Table 7: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Encryption with AES	BC-UnAuth	SP 800-38A and SP 800-38E. Encryption		AES-CBC: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS1: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS2: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS3: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CFB1: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CFB128: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CFB8: (A5747,

Name	Type	Description	Properties	Algorithms
				A5748, A5749, A5774, A5775, A5776, A5927) AES-CTR: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-ECB: (A5747, A5748, A5749, A5751, A5752, A5753, A5754, A5755, A5774, A5775, A5776, A5927, A5932) AES-OFB: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-XTS Testing Revision 2.0: (A5747, A5748, A5749, A5774, A5775, A5776, A5927)
Decryption with AES	BC-UnAuth	SP 800-38A and SP 800-38E. Decryption		AES-CBC: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS1: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS2: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CBC-CS3: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CFB1: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CFB128: (A5747, A5748, A5749, A5774, A5775, A5776, A5927)

Name	Type	Description	Properties	Algorithms
				AES-CFB8: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-CTR: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-ECB: (A5747, A5748, A5749, A5751, A5752, A5753, A5754, A5755, A5774, A5775, A5776, A5927, A5932) AES-OFB: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-XTS Testing Revision 2.0: (A5747, A5748, A5749, A5774, A5775, A5776, A5927)
Authenticated Encryption with AES	BC-Auth	SP 800-38D. Authenticated encryption		AES-CCM: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-GCM: (A5759, A5760, A5761, A5762, A5763, A5764, A5765, A5766, A5767, A5777, A5781, A5782, A5783, A5928, A5929, A5930) AES-KW: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-KWP: (A5747, A5748, A5749, A5774, A5775, A5776, A5927)
Authenticated Decryption with AES	BC-Auth	SP 800-38D. Authenticated decryption		AES-CCM: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-GCM: (A5759, A5760, A5761,

Name	Type	Description	Properties	Algorithms
				A5762, A5763, A5764, A5765, A5766, A5767, A5777, A5781, A5782, A5783, A5928, A5929, A5930) AES-KW: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-KWP: (A5747, A5748, A5749, A5774, A5775, A5776, A5927)
Message Authentication Generation with AES	MAC	SP 800-38B and SP 800-38D Message authentication generation		AES-CMAC: (A5747, A5748, A5749, A5774, A5775, A5776, A5927) AES-GMAC: (A5759, A5760, A5761, A5762, A5763, A5764, A5765, A5766, A5767, A5777, A5781, A5782, A5783, A5928, A5929, A5930)
Message Authentication Generation with HMAC	MAC	FIPS 198-1. Message authentication generation		HMAC-SHA-1: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) HMAC-SHA2-224: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) HMAC-SHA2-256: (A5750, A5768, A5769, A5770, A5771, A5778, A5779, A5931) HMAC-SHA2-384: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) HMAC-SHA2-512: (A5750, A5768, A5769, A5770, A5771, A5778,

Name	Type	Description	Properties	Algorithms
				A5931) HMAC-SHA2-512/224: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) HMAC-SHA2-512/256: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) HMAC-SHA3-224: (A5756, A5780, A5933) HMAC-SHA3-256: (A5756, A5780, A5933) HMAC-SHA3-384: (A5756, A5780, A5933) HMAC-SHA3-512: (A5756, A5780, A5933)
Message Authentication Generation with KMAC	MAC	SP 800-185. Message authentication generation		KMAC-128: (A5756, A5780, A5933) KMAC-256: (A5756, A5780, A5933)
Random Number Generation with DRBG	DRBG	SP 800-90ARev1. Random number generation		Counter DRBG: (A5746) Hash DRBG: (A5746) HMAC DRBG: (A5746)
Signature Generation with ECDSA	DigSig-SigGen	FIPS 186-5. Signature generation		ECDSA SigGen (FIPS186-5): (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)
Signature Generation with RSA	DigSig-SigGen	FIPS 186-5. Signature generation	IG C.F:RSA SigGen was CAVP tested with moduli sizes 2048, 3072, 4096 bits. The module supports moduli sizes larger than 4096 bits, up to 16384 bits.	RSA SigGen (FIPS186-5): (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)

Name	Type	Description	Properties	Algorithms
Signature Verification with ECDSA	DigSig-SigVer	FIPS 186-5. Signature verification		ECDSA SigVer (FIPS186-5): (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)
Signature Verification with RSA	DigSig-SigVer	FIPS 186-5. Signature verification	IG C.F:RSA SigVer was CAVP tested with moduli sizes 2048, 3072, 4096 bits. The module supports moduli sizes larger than 4096 bits, up to 16384 bits.	RSA SigVer (FIPS186-5): (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)
Key Pair Generation with ECDSA	AsymKeyPair-KeyGen CKG	FIPS 186-5. Key pair generation		ECDSA KeyGen (FIPS186-5): (A5745, A5750, A5768, A5769, A5770, A5771, A5778, A5931) Asymmetric Cryptographic Key Generation (CKG): ()
Key Pair Generation with RSA	AsymKeyPair-KeyGen CKG	FIPS 186-5. Key pair generation	IG C.F:RSA KeyGen was CAVP tested with moduli sizes of 2048, 3072, 4096, 6144, 8192 bits. The module supports moduli sizes larger than 8192 bits, up to 16384 bits. The number of Miller-Rabin tests is compliant with Table B.1 of FIPS 186-5.	RSA KeyGen (FIPS186-5): (A5750, A5768, A5769, A5770, A5771, A5778, A5931) Asymmetric Cryptographic Key Generation (CKG): ()
Key Pair Generation with Safe Primes	AsymKeyPair-KeyGen CKG	SP 800-56Ar3. Key pair generation		Safe Primes Key Generation: (A5773) Asymmetric Cryptographic Key Generation (CKG): ()
Key Pair Verification with ECDSA	AsymKeyPair-KeyVer	FIPS 186-5 and FIPS186-4. Key verification		ECDSA KeyVer (FIPS186-4): (A5745, A5750, A5768, A5769, A5770, A5771,

Name	Type	Description	Properties	Algorithms
				A5778, A5931) ECDSA KeyVer (FIPS186-5): (A5745, A5750, A5768, A5769, A5770, A5771, A5778, A5931)
Key Pair Verification with Safe Primes	AsymKeyPair-KeyVer	SP 800-56Ar3. Key pair verification		Safe Primes Key Verification: (A5773)
Key Derivation with KBKDF	KBKDF	SP 800-108r1. Key derivation		KDF SP800-108: (A5772)
Key Derivation with KDA OneStep	KAS-56CKDF	SP 800-56Cr2. Key derivation		KDA OneStep SP800-56Cr2: (A5744)
Key Derivation with KDA TwoStep	KAS-56CKDF	SP 800-56Cr2. Key derivation		KDA TwoStep SP800-56Cr2: (A5744)
Key Derivation with KDA HKDF	KAS-56CKDF	SP 800-56Cr2. Key derivation		KDA HKDF SP800-56Cr2: (A5758)
Key Derivation with ANS X9.42 KDF	KAS-135KDF	SP 800-135r1. Key derivation		KDF ANS 9.42: (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)
Key Derivation with X9.63 KDF	KAS-135KDF	SP 800-135r1. Key derivation		KDF ANS 9.63: (A5750, A5768, A5769, A5770, A5771, A5778, A5931)
Key Derivation with SSH KDF	KAS-135KDF	SP 800-135r1. Key derivation		KDF SSH: (A5751, A5752, A5753, A5754, A5755, A5932)
Key Derivation with TLS 1.2 KDF	KAS-135KDF	SP 800-135r1. Key derivation		TLS v1.2 KDF RFC7627: (A5750, A5768, A5769, A5770, A5771, A5778, A5931)
Key Derivation with TLS 1.3 KDF	KAS-135KDF	RFC 8446. Key derivation		TLS v1.3 KDF: (A5758)
Key Derivation with PBKDF2	PBKDF	SP 800-132. Key derivation		PBKDF: (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933)

Name	Type	Description	Properties	Algorithms
Shared Secret Computation	KAS-SSC	SP 800-56Ar3. Shared secret computation		KAS-ECC-SSC Sp800-56Ar3: (A5750, A5768, A5770, A5771, A5778, A5931, A5769) KAS-FFC-SSC Sp800-56Ar3: (A5773)
Message Digest with SHA	SHA	FIPS 180-4 and FIPS 202. Message digest		SHA-1: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) SHA2-224: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) SHA2-256: (A5750, A5768, A5769, A5770, A5771, A5778, A5779, A5931) SHA2-384: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) SHA2-512: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) SHA2-512/224: (A5750, A5768, A5769, A5770, A5771, A5778, A5931) SHA2-512/256: (A5750, A5771, A5778, A5931, A5768, A5769, A5770) SHA3-224: (A5756, A5780, A5933) SHA3-256: (A5756, A5780, A5933) SHA3-384: (A5756, A5780, A5933) SHA3-512: (A5756, A5780, A5933)
Message Digest with SHAKE	XOF	FIPS 202. Message digest		SHAKE-128: (A5756, A5780, A5933) SHAKE-256: (A5756, A5780, A5933)

Name	Type	Description	Properties	Algorithms
Signature Verification with RSA (Legacy)	DigSig-SigVer	FIPS 186-4. Legacy digital signature verification	IG C.M:Legacy Algorithms	RSA SigVer (FIPS186-4): (A5750, A5768, A5770, A5771, A5778, A5931) Modulo: 1024 Hash Algorithm: SHA-1 RSA SigVer (FIPS186-4): (A5769) Modulo : 1024 Hash Algorithm: SHA-1
Signature Verification with ECDSA (Legacy)	DigSig-SigVer	FIPS 186-4. Legacy digital signature verification	IG C.M:Legacy Algorithms	ECDSA SigVer (FIPS186-4): (A5750, A5756, A5768, A5769, A5770, A5771, A5778, A5780, A5931, A5933) Curve: P-192 Hash Algorithm: SHA-1

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

For TLS 1.2, the module offers the AES GCM implementation and uses the context of Scenario 1 of FIPS 140-3 IG C.H. The module is compliant with SP 800-52r2 Section 3.3.1 and the mechanism for IV generation is compliant with RFC 5288 and 8446.

The module does not implement the TLS protocol. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS 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.

Alternatively, the Crypto Officer can use the module's API to perform AES GCM encryption using internal IV generation. These IVs are always 96 bits and generated using the approved DRBG internal to the module's boundary in compliance with Scenario 2 of IG C.H.

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. The service can be requested by invoking the EVP_EncryptInit_ex2 API function with a non-NULL iv value. When this is the case, the API will set a non-approved service indicator as described in Section 4.3.

Finally, for TLS 1.3, the AES GCM implementation uses the context of Scenario 5 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher-suites that explicitly select AES GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES GCM cipher suites from Section 3.3.1 of SP800-52r2. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS 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

2.7.2 AES XTS

In accordance to FIPS 140-3 IG C.I, the module implements a check that ensures, before performing any cryptographic operation, that the two AES keys used in AES XTS mode are not identical. As the module does not generate symmetric keys, the check is performed when keys are input the service APIs.

Key_1 and Key_2 shall be generated and/or established independently according to the rules for component symmetric keys from NIST SP 800-133rev2, Sec. 6.3.

In addition, Section 4 of SP 800-38E states that 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.

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

2.7.3 Key Derivation using SP 800-132 PBKDF2

The module provides password-based key derivation (PBKDF2), compliant with SP 800-132. The module supports option 1a from Section 5.4 of SP 800-132, in which the Master Key (MK) or a segment of it is used directly as the Data Protection Key (DPK). In accordance to SP 800-132 and FIPS 140-3 IG D.N, the following requirements shall be met:

- Derived keys shall only be used in storage applications. The MK shall not be used for other purposes. The module accepts a minimum length of 112 bits for the MK or DPK .
- Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 8 characters. Assuming the worst-case scenario of all digits, this results in the estimated probability of guessing the password to be at most 10^{-8}. Combined with

the minimum iteration count as described below, this provides an acceptable trade-off between user experience and security against brute-force attacks.
- A portion of the salt, with a length of at least 128 bits (this is verified by the module to determine the service is approved), shall be generated randomly using the SP 800-90Ar1 DRBG provided by the module.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The module enforces the iteration count to be equal to or greater than 1000.

2.7.4 Compliance to SP 800-56Arev3 Assurances

The module offers DH and ECDH shared secret computation services compliant to the SP 800-56ARev3 and meeting IG D.F scenario 2 path (1). In order to meet the required assurances listed in section 5.6 of SP 800-56Arev3, the module shall be used together with an application that implements the “TLS protocol” and the following steps shall be performed.

1. The entity using the module, must use the module's "Key pair generation" service for generating DH/ECDH ephemeral keys. This meets the assurances required by key pair owner defined in the section 5.6.2.1 of SP 800-56ARev3.
2. As part of the module's shared secret computation (SSC) service, the module internally performs the public key validation on the peer's public key passed in as input to the SSC function. This meets the public key validity assurance required by the sections 5.6.2.2.2 of SP 800-56ARev3.
3. The module does not support static keys therefore the "assurance of peer's possession of private key" is not applicable.

2.7.5 Authenticated Encryption/Decryption

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

2.7.6 KAS-SSC

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.

2.8 RBG and Entropy

Cert Number	Vendor Name
E218	Canonical

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
OpenSSL CPU Time Jitter	Non-Physical	Ubuntu 24.04 on Supermicro SYS-1019P-WTR on Intel Xeon Gold 6226; Ubuntu 24.04 on Amazon Web Services (AWS) c7g.metal on AWS Graviton3; Ubuntu 24.04 on IBM Telum on IBM z16	256	Full entropy	SHA3-256 (A5588); Counter DRBG (kernel) (A5588, A5591, A5592, A5593, A5594, A5595, A5599, A5600, A5601, A5602, A5603, A5606, A5607, A5608, A5609, A5610, A5611); Counter DRBG (A5746)

Table 10: Entropy Sources

The module implements a primary DRBG (AES-256-CTR-DRBG (5746)) which acts as the conditioning component for the entropy source mentioned in the above table. It is only used internally by the module to seed the secondary DRBGs which can be of type (CTR, Hash, HMAC). The module complies with the Public Use Document for ESV certificate E218 by reading entropy data from the EVP_RAND_generate() function of the primary DRBG, which corresponds to the GetEntropy() conceptual interface. The operational environment on the ESV certificate is identical to the operating system described in this document. There are no maintenance requirements for the entropy source.

As per the Public Use document of entropy certificate E218, the entropy source provides full entropy of 256 bits.

When the module needs random data for internal purposes it uses two separate instances of AES-256 CTR_DRBG DRBG based on use case. i.e., it uses the “private DRBG” accessed via RAND_priv_bytes () for asymmetric key generation, signature generation, or other SSP use cases and it uses the “public DRBG” accessed via RAND_bytes() when it needs to generate IV or other non-SSP use cases.

When an external caller needs the random data, they can access it via “Random Number Generation” service of the module and they have a choice between Hash, HMAC or CTR_DRBG listed in the Approved Algorithms table.

2.9 Key Generation

The module implements asymmetric key pair generation compliant with SP 800-133 Rev. 2 as listed in the Security Function Implementations table.

When random values are required, they are obtained from the SP 800-90A Rev. 1 approved DRBG, compliant with Section 4 of SP 800-133 Rev. 2 (without XOR). Intermediate key generation values are not output from the module and are explicitly zeroized after processing the service.

The key derivation methods implemented by the module are specified in the Security Function Implementations table.

2.10 Key Establishment

Key Establishment methods are specified in the Security Function Implementations table.

2.11 Industry Protocols

The module implements the SSH KDF (CVL) for use in the SSH protocol (RFC 4253 and RFC 6668).

GCM with internal IV generation in the approved mode is compliant with versions 1.2 and 1.3 of the TLS protocol (RFC 5288 and 8446) and shall only be used in conjunction with the TLS protocol. Additionally, the module implements the TLS 1.2 and TLS 1.3 key derivation functions for use in the TLS protocol.

For Diffie-Hellman, the module supports the use of the safe primes defined in RFC 3526 (IKE) and RFC 7919 (TLS). Note that the module only implements key pair generation, key pair verification, and shared secret computation. No other part of the IKE or TLS protocols is implemented (with the exception of the TLS 1.2 KDF (CVL) and 1.3 KDF (CVL)):

- IKE (RFC 3526): MODP-2048 (ID = 14), MODP-3072 (ID = 15), MODP-4096 (ID = 16), MODP-6144 (ID = 17), MODP-8192 (ID = 18)
- TLS (RFC 7919): ffdhe2048 (ID = 256), ffdhe3072 (ID = 257), ffdhe4096 (ID = 258), ffdhe6144 (ID = 259), ffdhe8192 (ID = 260)

For Elliptic Curve Diffie-Hellman, the module supports the NIST-defined P-224, P-256, P-384, and P-521 curves.

No parts of the SSH, TLS, or IKE protocols, other than those mentioned above, have been tested by the CAVP or CMVP.

2.12 Additional Information

Not applicable.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API input parameters
N/A	Data Output	API output parameters
N/A	Control Input	API function calls
N/A	Status Output	API return codes, error queue

Table 11: Ports and Interfaces

As a software-only module, the module does not have physical ports. Physical Ports are interpreted to be the physical ports of the hardware platform on which it runs. The module does not implement a control output interface.

3.2 Trusted Channel Specification

Not applicable.

3.3 Control Interface Not Inhibited

Not applicable.

3.4 Additional Information

Not applicable.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 12: Roles

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

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Digest	Compute a message digest	0	Message	Message digest	Message Digest with SHA Message Digest with SHAKE	Crypto Officer
Symmetric Encryption	Encrypt a plaintext	0	AES Key, plaintext, IV	Ciphertext	Encryption with AES	Crypto Officer - AES Key: W,E
Symmetric Decryption	Decrypt a ciphertext	0	AES Key, ciphertext, IV	Plaintext	Decryption with AES	Crypto Officer - AES Key: W,E
Authenticated Symmetric Encryption	Encrypt and authenticate a plaintext	0	AES Key, plaintext, IV	Ciphertext, MAC tag	Authenticated Encryption with AES	Crypto Officer - AES Key: W,E
Authenticated Symmetric Decryption	Decrypt and authenticate a ciphertext	0	AES Key, ciphertext, MAC tag, IV	Plaintext or Failure	Authenticated Decryption with AES	Crypto Officer - AES Key: W,E
AES Message Authentication Generation	Compute a MAC tag using AES	0	AES Key, message	MAC tag	Message Authentication Generation with AES	Crypto Officer - AES Key: W,E
HMAC Message Authentication Generation	Compute a MAC tag using HMAC	0	HMAC Key, message	MAC tag	Message Authentication Generation with HMAC	Crypto Officer - HMAC Key: W,E
KMAC Message Authentication Generation	Compute a MAC tag using KMAC	0	KMAC Key, message	MAC tag	Message Authentication Generation with KMAC	Crypto Officer - KMAC Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
TLS KDF Key Derivation	TLS key derivation	0	Shared Secret	TLS Derived Key	Key Derivation with TLS 1.2 KDF Key Derivation with TLS 1.3 KDF	Crypto Officer - Shared Secret: W,E - TLS Derived Key: G,R
KBKDF Key Derivation	Derive a key from a key-derivation key	0	Key-Derivation Key	KBKDF Derived Key	Key Derivation with KBKDF	Crypto Officer - Key-Derivation Key: W,E - KBKDF Derived Key: G,R
ANS X9.42 Key Derivation	Derive a key from a shared secret	0	Shared Secret	ANS X9.42 Derived Key	Key Derivation with ANS X9.42 KDF	Crypto Officer - ANS X9.42 Derived Key: G,R - Shared Secret: W,E
ANS X9.63 Key Derivation	Derive a key from a shared secret	0	Shared Secret	ANS X9.63 Derived Key	Key Derivation with X9.63 KDF	Crypto Officer - Shared Secret: W,E - ANS X9.63 Derived Key: G,R
HKDF Key Derivation	Derive a key from a shared secret	0	Shared Secret	HKDF Derived key	Key Derivation with KDA HKDF	Crypto Officer - Shared Secret: W,E - HKDF Derived Key: G,R
OneStep KDA Key Derivation	Derive a key from a shared secret	0	Shared Secret	KDA OneStep Derived Key	Key Derivation with KDA OneStep	Crypto Officer - Shared Secret: W,E - KDA OneStep Derived Key: G,R
TwoStep KDA Key Derivation	Derive a key from a shared secret	0	Shared Secret	KDA TwoStep Derived Key	Key Derivation with KDA TwoStep	Crypto Officer - Shared Secret: W,E - KDA TwoStep Derived Key: G,R
SSH KDF key derivation	Derive a key from a shared secret	0	Shared Secret	SSH KDF Derived Key	Key Derivation with SSH KDF	Crypto Officer - Shared Secret: W,E - SSH KDF

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Derived Key: G,R
PBKDF Key Derivation	Derive a key from a password	0	Password	PBKDF Derived Key	Key Derivation with PBKDF2	Crypto Officer - Password: W,E - PBKDF Derived Key: G,R
Random Number Generation	Generate random number	0	Number of bits	Random number	Random Number Generation with DRBG	Crypto Officer - Entropy Input: W,E - DRBG Seed: G,E - DRBG Internal State (V, Key): G,E - DRBG Internal State (V, C): G,E
KAS-FFC-SSC Shared Secret Computation	Compute a shared secret	0	DH Private Key (owner), DH Public Key (peer)	Shared Secret	Shared Secret Computation	Crypto Officer - Shared Secret: G,R - DH Private Key: W,E - DH Public Key: W,E
KAS-ECC-SSC Shared Secret Computation	Compute a shared secret	0	EC Private Key (owner), EC Public Key (peer)	Shared Secret	Shared Secret Computation	Crypto Officer - Shared Secret: G,R - EC Private Key: W,E - EC Public Key: W,E
RSA Digital Signature Generation	Generate a digital signature with RSA	0	RSA Private Key, message, hash algorithm	Signature	Signature Generation with RSA	Crypto Officer - RSA Private Key: W,E
ECDSA Digital Signature Generation	Generate a digital signature with ECDSA	0	EC Private Key, message, hash algorithm	Signature	Signature Generation with ECDSA	Crypto Officer - EC Private Key: W,E
RSA Digital Signature Verification	Verify a digital signature using RSA	0	RSA Public Key, message, signature, hash algorithm	Pass or Fail	Signature Verification with RSA Signature Verification with RSA (Legacy)	Crypto Officer - RSA Public Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
ECDSA Digital Signature Verification	Verify a digital signature using ECDSA	0	EC Public Key, message, signature, hash algorithm	Pass or Fail	Signature Verification with ECDSA Signature Verification with ECDSA (Legacy)	Crypto Officer - EC Public Key: W,E
RSA Key Pair Generation	Generate an RSA key pair	0	Modulus bits	Module Generated RSA Private Key, Module Generated RSA Public Key	Key Pair Generation with RSA	Crypto Officer - Module Generated RSA Private Key: G,R - Module Generated RSA Public Key: G,R - Intermediate Key Generation Value: G,E,Z
ECDSA Key Pair Generation	Generate an EC key pair	0	Curve	Module Generated EC Private Key, Module Generated EC Public Key	Key Pair Generation with ECDSA	Crypto Officer - Module Generated EC Private Key: G,R - Module Generated EC Public Key: G,R - Intermediate Key Generation Value: G,E,Z
Safe Primes Key Pair Generation	Generate an DH key pair	0	Group	Module Generated DH Private Key, Module Generated DH Public Key	Key Pair Generation with Safe Primes	Crypto Officer - Module Generated DH Private Key: G,R - Module Generated DH Public Key: G,R - Intermediate Key Generation Value: G,E,Z
ECDSA Key Pair Verification	Verify an EC key pair	0	EC Private Key, EC Public Key	Pass or Fail	Key Pair Verification with ECDSA	Crypto Officer - EC Private Key: W,E - EC Public Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Safe Prime Key Pair Verification	Verify a DH key pair	0	DH Private Key, DH Public Key	Pass or Fail	Key Pair Verification with Safe Primes	Crypto Officer - DH Private Key: W,E - DH Public Key: W,E
Show Version	Return the name and version information	0	None	Module name and version	None	Crypto Officer
Show Status	Return the module status	0	None	Module status	None	Crypto Officer
Self-Test	Perform the CASTs and integrity test	0	None	Pass or Fail of self-tests	None	Crypto Officer
Zeroization	Zeroize any SSP	0	An SSP	None	None	Crypto Officer - AES Key: Z - HMAC Key: Z - KMAC Key: Z - Key-Derivation Key: Z - Shared Secret: Z - Password: Z - PBKDF Derived Key: Z - KBKDF Derived Key: Z - ANS X9.42 Derived Key: Z - ANS X9.63 Derived Key: Z - HKDF Derived Key: Z - KDA OneStep Derived Key: Z - KDA TwoStep Derived Key: Z - TLS Derived Key: Z - SSH KDF Derived Key: Z - Entropy Input: Z - DRBG Internal State (V, Key): Z - DRBG Seed:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Z - DH Private Key: Z - DH Public Key: Z - EC Private Key: Z - EC Public Key: Z - RSA Private Key: Z - RSA Public Key: Z - Module Generated DH Private Key: Z - Module Generated DH Public Key: Z - Module Generated EC Private Key: Z - Module Generated EC Public Key: - Module Generated RSA Private Key: Z - Module Generated RSA Public Key: Z

Table 13: Approved Services

The module provides services to operators that assume the available role. All services are described in detail in the API documentation (manual pages). The Approved Services table and the Non-Approved Services table define the services that utilize approved and non-approved security functions in this module. For the respective tables, the convention below applies when specifying the access permissions (types) that the service has for each SSP.

- **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.

To interact with the module, a calling application must use the EVP API layer provided by OpenSSL. This layer will delegate the request to the FIPS provider, which will in

turn perform the requested service. Additionally, this EVP API layer can be used to retrieve the approved service indicator for the module.

The cryptographic module provides an approved service indicator in the form of an OpenSSL provider gettable parameter called UBUNTU_OSSL_PROV_FIPS_PARAM_UNAPPROVED_USAGE. This parameter will be equal to 0 if the requested service is an approved security service, otherwise it will be set to 1. The operator is responsible to query the value of such gettable parameter after calling the requested service.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Encryption	AES GCM (external IV)	AES GCM (external IV)	CO
Key Pair Generation	Key pair generation	DSA ECDSA with curve P-192, B-163, K-163	CO
Key Pair Verification	Key pair verification	DSA ECDSA with curve B-163, K-163	CO
Signature Generation	Signature generation	DSA ECDSA with curve P-192 ECDSA with curve B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571 RSA and ECDSA (pre-hashed message) RSA X9.31	CO
Signature Verification	Signature verification	DSA ECDSA with curve B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571 RSA and ECDSA (pre-hashed message) RSA X9.31	CO
Asymmetric Encryption	Asymmetric encryption	RSA primitive RSA-OAEP	CO
Asymmetric Decryption	Asymmetric decryption	RSA primitive RSA-OAEP	CO
Secret Value Encapsulation	Secret value encapsulation	RSASVE	CO
Secret Value Un-encapsulation	Secret value un-encapsulation	RSASVE	CO

Table 14: Non-Approved Services

In the table above, CO specifies the Crypto Officer role.

4.5 External Software/Firmware Loaded

The module does not have the capability of loading software or firmware from an external source.

4.6 Bypass Actions and Status

Not applicable.

4.7 Cryptographic Output Actions and Status

Not applicable.

4.8 Additional Information

Not applicable.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC-SHA2-256 value calculated at run time with the HMAC-SHA2-256 value embedded in the fips.so file that was computed at build time.

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 test may be invoked on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. This will perform (among others) the software integrity test.

5.3 Open-Source Parameters

Not applicable.

5.4 Additional Information

Not applicable.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

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

6.2 Configuration Settings and Restrictions

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

6.3 Additional Information

There are no concurrent operators.

7 Physical Security

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

8 Non-Invasive Security

8.1 Mitigation Techniques

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

8.2 Effectiveness

Not applicable.

8.3 Additional Information

Not applicable.

9 Sensitive Security Parameters Management

9.1 Storage Areas

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

Table 15: Storage Areas

SSPs are provided to the module by the calling application and are destroyed when released by the appropriate API function calls. The module does not perform persistent storage of SSPs.

9.2 SSP Input-Output Methods

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

Table 16: SSP Input-Output Methods

The module only supports SSP entry and output to and from the calling application running on the same operational environment. This corresponds to manual distribution, electronic entry/output (“CM Software to/from App via TOEPP Path”) per FIPS 140-3 IG 9.5.A Table 1. There is no entry or output of cryptographically protected SSPs.

SSPs can be entered into the module via API input parameters, when required by a service. SSPs can also be output from the module via API output parameters, immediately after generation of the SSP.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the cipher handle: EVP_CIPHER_CTX_free() clears and frees symmetric cipher context, EVP_MAC_CTX_free() clears and frees MAC context, EVP_KDF_CTX_free() clears and frees KDF context, EVP_RAND_CTX_free() clears and frees DRBG context, EVP_PKEY_free() clears and frees asymmetric key pair structures	Memory occupied by SSPs is overwritten with zeroes and then it is released, which renders the SSP values irretrievable. The completion of the zeroization routine indicates that the	By calling the cipher related zeroization API

Zeroization Method	Description	Rationale	Operator Initiation
		zeroization procedure succeeded.	
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A
Module reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 17: SSP Zeroization Methods

The memory occupied by SSPs is allocated by regular memory allocation operating system calls. The operator is responsible for calling the appropriate destruction functions provided in the module's API. The destruction functions, listed above, overwrite the memory occupied by SSPs with zeroes and de-allocate the memory with the regular memory de-allocation operating system call. All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	Used for encryption, decryption, and message authentication	128, 192, 256 bits - 128, 192, 256 bits	Symmetric key - CSP			Encryption with AES Decryption with AES Authenticated Encryption with AES Authenticated Decryption with AES Message Authentication Generation with AES
HMAC Key	Used for hash-based message authentication	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Generation with HMAC
KMAC Key	Used for message authentication	128-1024 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Generation with KMAC
Key-Derivation Key	Used for key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key Derivation with KBKDF

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Shared Secret	Generated by shared secret computation and used for key derivation	224-8192 bits - 112-256 bits	Shared secret - CSP		Shared Secret Computation	Key Derivation with KDA OneStep Key Derivation with KDA TwoStep Key Derivation with KDA HKDF Key Derivation with ANS X9.42 KDF Key Derivation with X9.63 KDF Key Derivation with SSH KDF Key Derivation with TLS 1.2 KDF Key Derivation with TLS 1.3 KDF
Password	Used for password-based key derivation	At least 8 characters - N/A	Password - CSP			Key Derivation with PBKDF2
PBKDF Derived Key	Generated by password-based key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with PBKDF2		
KBKDF Derived Key	Generated by key-based key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KBKDF		
ANS X9.42 Derived Key	Generated by ANS X9.42 key derivation	128-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with ANS X9.42 KDF		
ANS X9.63 Derived Key	Generated by ANS X9.63 key derivation	128-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with X9.63 KDF		
HKDF Derived Key	Generated by HKDF key derivation	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KDA HKDF		
KDA OneStep Derived Key	Generated by OneStep KDA key derivation	2048 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KDA OneStep		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KDA TwoStep Derived Key	Generated by TwoStep KDA key derivation	2048 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KDA TwoStep		
TLS Derived Key	Generated by TLS KDF key derivation	112-1024 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with TLS 1.2 KDF Key Derivation with TLS 1.3 KDF		
SSH KDF Derived Key	Generated by SSH KDF key derivation	112-256 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with SSH KDF		
Entropy Input	Used for random number generation and seeding a DRBG (compliant with IG D.L)	128-384 bits - 128-256 bits	Entropy input - CSP			Random Number Generation with DRBG
DRBG Internal State (V, Key)	Used for random number generation (compliant with IG D.L)	Counter DRBG: 256, 320, 348 bits; HMAC DRBG: 320, 512, 1024 bits - Counter DRBG: 128, 192, 256 bits; HMAC DRBG: 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
DRBG Internal State (V, C)	Used for random number generation (compliant with IG D.L)	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
DRBG Seed	Used for random number generation (compliant with IG D.L)	128-256 bits - 128-256 bits	Seed - CSP	Random Number Generation with DRBG		Random Number Generation with DRBG
DH Private Key	Used for shared secret computation and key pair verification	2048-8192 bits - 112-200 bits	Private key - CSP			Key Pair Verification with Safe Primes Shared Secret Computation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DH Public Key	Used for shared secret computation and key pair verification	2048-8192 bits - 112-200 bits	Public key - PSP			Key Pair Verification with Safe Primes Shared Secret Computation
EC Private Key	Used for shared secret computation, digital signature generation, and key pair verification	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 bits - 112-256 bits	Private key - CSP			Signature Generation with ECDSA Key Pair Verification with ECDSA Shared Secret Computation
EC Public Key	Used for shared secret computation, signature verification, and key pair verification	P-192, P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409, K-571, B-163, B-233, B-283, B-409, B-571 bits - 80-256 bits	Public key - PSP			Signature Verification with ECDSA Key Pair Verification with ECDSA Shared Secret Computation
RSA Private Key	Used for signature generation	2048-16384 bits - 112-256 bits	Private key - CSP			Signature Generation with RSA
RSA Public Key	Used for signature verification	1024-16384 bits - 80-256 bits	Public key - PSP			Signature Verification with RSA
Module Generated DH Private Key	DH private key generated by the module	2048-8192 bits - 112-200 bits	Private key - CSP	Key Pair Generation with Safe Primes		
Module Generated DH Public Key	DH public key generated by the module	2048-8192 bits - 112-200 bits	Public key - PSP	Key Pair Generation with Safe Primes		
Module Generated EC Private Key	EC private key generated by the module	P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, B-571 bits - 112-256 bits	Private key - CSP	Key Pair Generation with ECDSA		
Module Generated EC Public Key	EC public key generated by the module	P-224, P-256, P-384, P-521, K-163, K-233, K-283, K-409,	Public key - PSP	Key Pair Generation with ECDSA		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		K-571, B-163, B-233, B-283, B-409, B-571 bits - 112-256 bits				
Module Generated RSA Private Key	RSA private key generated by the module	2048-16384 bits - 112-256 bits	Private key - CSP	Key Pair Generation with RSA		
Module Generated RSA Public Key	RSA public key generated by the module	2048-16384 bits - 112-256 bits	Public key - PSP	Key Pair Generation with RSA		
Intermediate Key Generation Value	Used for key pair generation	224-16384 bits - 112-256 bits	Intermediate value - CSP	Key Pair Generation with ECDSA Key Pair Generation with RSA Key Pair Generation with Safe Primes		Key Pair Generation with ECDSA Key Pair Generation with RSA Key Pair Generation with Safe Primes

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	
HMAC Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	
KMAC Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	
Key-Derivation Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	KBKDF Derived Key:Derives
Shared Secret	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	DH Private Key:Generated From DH Public Key:Generated From EC Private

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					Key:Generated From EC Public Key:Generated From
Password	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	PBKDF Derived Key:Derives
PBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Password:Derived From
KBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Key-Derivation Key:Derived From
ANS X9.42 Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
ANS X9.63 Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
HKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
KDA OneStep Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
KDA TwoStep Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
TLS Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
SSH KDF Derived Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Shared Secret:Derived From
Entropy Input		RAM:Plaintext	From service invocation to service completion	Automatic	DRBG Seed:Generates
DRBG Internal State (V, Key)		RAM:Plaintext	From DRBG instantiation to un-	Free cipher handle	DRBG Seed:Generated From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			instantiation or internal zeroization	Module reset	
DRBG Internal State (V, C)		RAM:Plaintext	From DRBG instantiation to un-instantiation or internal zeroization	Free cipher handle Module reset	DRBG Seed:Generated From
DRBG Seed		RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	DRBG Internal State (V, Key):Generates DRBG Internal State (V, C):Generates Entropy Input:Generated From
DH Private Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	DH Public Key:Paired With Shared Secret:Derives
DH Public Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	DH Private Key:Paired With Shared Secret:Generates
EC Private Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	EC Public Key:Paired With Shared Secret:Generates
EC Public Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	EC Private Key:Paired With Shared Secret:Generates
RSA Private Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	RSA Public Key:Paired With
RSA Public Key	API input parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	RSA Private Key:Paired With
Module Generated DH Private Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated DH Public Key:Paired With Intermediate Key Generation Value:Generated From
Module Generated DH Public Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated DH Private Key:Paired With Intermediate Key Generation Value:Generated From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Module Generated EC Private Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated EC Public Key:Paired With Intermediate Key Generation Value:Generated From
Module Generated EC Public Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated EC Private Key:Paired With Intermediate Key Generation Value:Generated From
Module Generated RSA Private Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated RSA Public Key:Paired With Intermediate Key Generation Value:Generated From
Module Generated RSA Public Key	API output parameters	RAM:Plaintext	From service invocation to service completion	Free cipher handle Module reset	Module Generated RSA Private Key:Paired With Intermediate Key Generation Value:Generated From
Intermediate Key Generation Value		RAM:Plaintext	From service invocation to service completion	Automatic	Module Generated DH Private Key:Generates Module Generated DH Public Key:Generates Module Generated EC Private Key:Generates Module Generated EC Public Key:Generates Module Generated RSA Private Key:Generates Module Generated RSA Public Key:Generates

Table 19: SSP Table 2

The tables above summarize the Sensitive Security Parameters (SSPs) that are used by the cryptographic services implemented in the module in the approved services (Approved Services table).

SSPs, including CSPs, are directly imported as input parameters and exported as output parameters from the module. Because these SSPs are only transiently used for a specific service, they are, by definition, exclusive between approved and non-approved services.

9.5 Transitions

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

9.6 Additional Information

Not applicable.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A5750)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5768)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5769)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5770)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5771)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5778)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5779)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so
HMAC-SHA2-256 (A5931)	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so

Table 20: Pre-Operational Self-Tests

The module performs pre-operational tests automatically when the module is powered on. The pre-operational self-tests ensure that the module is not corrupted. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

The integrity of the shared library component of the module is verified by comparing an HMAC-SHA2-256 value calculated at run time with the corresponding HMAC value embedded in the fips.so file that was computed at build time.

If the software integrity test fails, the module transitions to the error state (Section 10.3). The HMAC and SHA2-256 algorithms go through their respective CASTs before the software integrity test is performed.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A5747)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5748)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5749)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5751)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5752)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5753)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5754)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5755)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5774)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5775)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5776)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A5927)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and	Symmetric operation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				services are available for use		before the integrity test
AES-ECB (A5932)	Decrypt with 256-bit key	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5759)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5760)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5761)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5762)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5763)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5764)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5765)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5766)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5767)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5777)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A5781)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5782)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5783)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5928)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5929)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A5930)	Encrypt/Decrypt with 256-bit key, 96-bit (internal IV)	KAT	CAST	Module becomes operational and services are available for use	Symmetric operation	Test runs at power-on before the integrity test
SHA-1 (A5750)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA-1 (A5768)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA-1 (A5769)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA-1 (A5770)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA-1 (A5771)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA-1 (A5778)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	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 (A5931)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5750)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5768)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5769)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5770)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5771)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5778)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A5931)	3-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A5756)	4-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A5780)	4-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A5933)	4-byte message	KAT	CAST	Module becomes operational and services are available for use	Message digest	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5750)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-256 (A5768)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5769)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5770)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5771)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5778)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5779)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-256 (A5931)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Message authentication	Test runs at power-on before the integrity test
Counter DRBG (A5746)	128 bit keys, DF, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC DRBG (A5746)	HMAC-SHA-1, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
Hash DRBG (A5746)	SHA2-256, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5750)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5768)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 (A5769)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5770)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5771)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5778)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A5931)	P-256 curve	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A5773)	ffdhe2048	KAT	CAST	Module becomes operational and services are available for use	Shared secret computation	Test runs at power-on before the integrity test
KDF SP800-108 (A5772)	HMAC-SHA2-256 with 128-bit key, 24-bit salt	KAT	CAST	Module becomes operational and services are available for use	Key based key derivation	Test runs at power-on before the integrity test
KDA OneStep SP800-56Cr2 (A5744)	SHA2-224	KAT	CAST	Module becomes operational and services are available for use	Shared secret key derivation	Test runs at power-on before the integrity test
KDA TwoStep SP800-56Cr2 (A5744)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Shared secret key derivation	Test runs on power-on before the integrity test
KDF ANS 9.42 (A5750)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5756)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5768)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF ANS 9.42 (A5769)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5770)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5771)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5778)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5780)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5931)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A5933)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5750)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5768)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5769)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5770)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5771)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF ANS 9.63 (A5778)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A5931)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.42 key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5750)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5768)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5769)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5770)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5771)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5778)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A5931)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.3 KDF (A5758)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based TLS v1.3 KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A5751)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A5752)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF SSH (A5753)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A5754)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A5755)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A5932)	SHA-1	KAT	CAST	Module becomes operational and services are available for use	Industry-based SSH KDF key derivation	Test runs at power-on before the integrity test
PBKDF (A5750)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5756)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5768)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5769)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5770)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5771)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5778)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5780)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
PBKDF (A5931)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
PBKDF (A5933)	SHA2-256 with 4096 iterations and 288-bit salt	KAT	CAST	Module becomes operational and services are available for use	Password-based key derivation	Test runs at power-on before the integrity test
KDA HKDF SP800-56Cr2 (A5758)	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Shared secret key derivation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5750)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5756)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5768)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5769)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5770)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5771)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5778)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5780)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-5) (A5931)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigGen (FIPS186-5) (A5933)	SHA2-256 with P-224, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5750)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5756)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5768)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5769)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5770)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5771)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5778)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5780)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5931)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-5) (A5933)	SHA2-256 with P-256, B-233	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5750)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigGen (FIPS186-5) (A5756)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5768)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5769)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5770)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5771)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5778)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5780)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5931)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-5) (A5933)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5750)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5756)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5768)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
RSA SigVer (FIPS186-5) (A5769)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5770)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5771)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5778)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5780)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5931)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-5) (A5933)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Digital signature verification	Test runs at power-on before the integrity test
ECDSA KeyGen (FIPS186-5) (A5745)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5750)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5768)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5769)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5770)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA KeyGen (FIPS186-5) (A5771)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5778)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
ECDSA KeyGen (FIPS186-5) (A5931)	SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5750)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5768)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5769)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5770)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5771)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5778)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5) (A5931)	PKCS#1 v1.5 with SHA2-256	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
Safe Primes Key Generation (A5773)	Section 5.6.2.1.4 of SP800-56Arev3	PCT	PCT	Successful key pair generation	SP 800-56ARev3, 5.6.2.1.4	Key pair generation

Table 21: Conditional Self-Tests

The module performs self-tests on all approved cryptographic algorithms as part of the approved services supported in the approved mode of operation, using the tests shown in the table above. The CASTs can be performed on demand by unloading and re-initializing the module. Data output through the data output interface is inhibited during the self-tests. If any of these tests fails, the module transitions to the error state.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A5750)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5768)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5769)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5770)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5771)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5778)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5779)	Message Authentication	SW/FW Integrity	On demand	Manually
HMAC-SHA2-256 (A5931)	Message Authentication	SW/FW Integrity	On demand	Manually

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A5747)	KAT	CAST	On Demand	Manually
AES-ECB (A5748)	KAT	CAST	On Demand	Manually
AES-ECB (A5749)	KAT	CAST	On Demand	Manually
AES-ECB (A5751)	KAT	CAST	On Demand	Manually
AES-ECB (A5752)	KAT	CAST	On Demand	Manually
AES-ECB (A5753)	KAT	CAST	On Demand	Manually
AES-ECB (A5754)	KAT	CAST	On Demand	Manually
AES-ECB (A5755)	KAT	CAST	On Demand	Manually
AES-ECB (A5774)	KAT	CAST	On Demand	Manually
AES-ECB (A5775)	KAT	CAST	On Demand	Manually
AES-ECB (A5776)	KAT	CAST	On Demand	Manually
AES-ECB (A5927)	KAT	CAST	On Demand	Manually
AES-ECB (A5932)	KAT	CAST	On Demand	Manually
AES-GCM (A5759)	KAT	CAST	On Demand	Manually
AES-GCM (A5760)	KAT	CAST	On Demand	Manually
AES-GCM (A5761)	KAT	CAST	On Demand	Manually
AES-GCM (A5762)	KAT	CAST	On Demand	Manually
AES-GCM (A5763)	KAT	CAST	On Demand	Manually
AES-GCM (A5764)	KAT	CAST	On Demand	Manually
AES-GCM (A5765)	KAT	CAST	On Demand	Manually
AES-GCM (A5766)	KAT	CAST	On Demand	Manually
AES-GCM (A5767)	KAT	CAST	On Demand	Manually
AES-GCM (A5777)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A5781)	KAT	CAST	On Demand	Manually
AES-GCM (A5782)	KAT	CAST	On Demand	Manually
AES-GCM (A5783)	KAT	CAST	On Demand	Manually
AES-GCM (A5928)	KAT	CAST	On Demand	Manually
AES-GCM (A5929)	KAT	CAST	On Demand	Manually
AES-GCM (A5930)	KAT	CAST	On Demand	Manually
SHA-1 (A5750)	KAT	CAST	On Demand	Manually
SHA-1 (A5768)	KAT	CAST	On Demand	Manually
SHA-1 (A5769)	KAT	CAST	On Demand	Manually
SHA-1 (A5770)	KAT	CAST	On Demand	Manually
SHA-1 (A5771)	KAT	CAST	On Demand	Manually
SHA-1 (A5778)	KAT	CAST	On Demand	Manually
SHA-1 (A5931)	KAT	CAST	On Demand	Manually
SHA2-512 (A5750)	KAT	CAST	On Demand	Manually
SHA2-512 (A5768)	KAT	CAST	On Demand	Manually
SHA2-512 (A5769)	KAT	CAST	On Demand	Manually
SHA2-512 (A5770)	KAT	CAST	On Demand	Manually
SHA2-512 (A5771)	KAT	CAST	On Demand	Manually
SHA2-512 (A5778)	KAT	CAST	On Demand	Manually
SHA2-512 (A5931)	KAT	CAST	On Demand	Manually
SHA3-256 (A5756)	KAT	CAST	On Demand	Manually
SHA3-256 (A5780)	KAT	CAST	On Demand	Manually
SHA3-256 (A5933)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5750)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5768)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5769)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5770)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5771)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5778)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5779)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A5931)	KAT	CAST	On Demand	Manually
Counter DRBG (A5746)	KAT	CAST	On Demand	Manually
HMAC DRBG (A5746)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Hash DRBG (A5746)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5750)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5768)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5769)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5770)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5771)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5778)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A5931)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A5773)	KAT	CAST	On Demand	Manually
KDF SP800-108 (A5772)	KAT	CAST	On Demand	Manually
KDA OneStep SP800-56Cr2 (A5744)	KAT	CAST	On Demand	Manually
KDA TwoStep SP800-56Cr2 (A5744)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5750)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5756)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5768)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5769)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5770)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5771)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5778)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDF ANS 9.42 (A5780)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5931)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A5933)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5750)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5768)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5769)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5770)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5771)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5778)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A5931)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5750)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5768)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5769)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5770)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5771)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5778)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A5931)	KAT	CAST	On Demand	Manually
TLS v1.3 KDF (A5758)	KAT	CAST	On Demand	Manually
KDF SSH (A5751)	KAT	CAST	On Demand	Manually
KDF SSH (A5752)	KAT	CAST	On Demand	Manually
KDF SSH (A5753)	KAT	CAST	On Demand	Manually
KDF SSH (A5754)	KAT	CAST	On Demand	Manually
KDF SSH (A5755)	KAT	CAST	On Demand	Manually
KDF SSH (A5932)	KAT	CAST	On Demand	Manually
PBKDF (A5750)	KAT	CAST	On Demand	Manually
PBKDF (A5756)	KAT	CAST	On Demand	Manually
PBKDF (A5768)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
PBKDF (A5769)	KAT	CAST	On Demand	Manually
PBKDF (A5770)	KAT	CAST	On Demand	Manually
PBKDF (A5771)	KAT	CAST	On Demand	Manually
PBKDF (A5778)	KAT	CAST	On Demand	Manually
PBKDF (A5780)	KAT	CAST	On Demand	Manually
PBKDF (A5931)	KAT	CAST	On Demand	Manually
PBKDF (A5933)	KAT	CAST	On Demand	Manually
KDA HKDF SP800-56Cr2 (A5758)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5750)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5756)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5768)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5769)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5770)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5771)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5778)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5780)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5931)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A5933)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5750)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5756)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5768)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5769)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5770)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5771)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5778)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5780)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-5) (A5931)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A5933)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5750)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5756)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5768)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5769)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5770)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5771)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5778)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5780)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5931)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A5933)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5750)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5756)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5768)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5769)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5770)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5771)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5778)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5780)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5931)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A5933)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5745)	PCT	PCT	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA KeyGen (FIPS186-5) (A5750)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5768)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5769)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5770)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5771)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5778)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-5) (A5931)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5750)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5768)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5769)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5770)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5771)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5778)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5) (A5931)	PCT	PCT	On Demand	Manually
Safe Primes Key Generation (A5773)	PCT	PCT	On Demand	Manually

Table 23: Conditional Periodic Information

The module does not implement periodic self-tests.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	The module immediately stops functioning	Software integrity test failure CAST failure PCT failure	Re-initialization of the module	Module will not load; Module is aborted for PCT failure

Table 24: Error States

If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process. Consequently, the data output interface is inhibited, and the module accepts no more inputs or requests (as the module is no longer running).

Regarding the PCT failure, an OSSL_PROV_PARAM_STATUS parameter can be queried from the FIPS provider to check the status of the cryptographic module.

The table above lists the error states and the status indicator values that explain the error that has occurred.

10.5 Operator Initiation of Self-Tests

The software integrity tests and cryptographic algorithm self-tests can be invoked on demand by resetting the module or by invoking the OSSL_PROVIDER_self_test method. The pair-wise consistency tests can be invoked on demand by requesting the key pair generation service.

10.6 Additional Information

Not applicable.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The binaries of the FIPS validated module are contained in the following Ubuntu packages for delivery:

- openssl-fips-module- 3.0.13-0ubuntu3+Fips1_amd64.deb for X86_64
- openssl-fips-module- 3.0.13-0ubuntu3+Fips1_arm64.deb for ARM64
- openssl-fips-module- 3.0.13-0ubuntu3+Fips1_s390x.deb for s390x

Once the operating environment is configured following the instructions provided, the Crypto Officer can install the Ubuntu packages containing the module listed below - Ubuntu packages using the Advanced Package Tool (APT) with the following command:

```
$ sudo apt-get install openssl-fips-module-3
```

All the Ubuntu packages are associated with hashes for integrity check. The integrity of the Ubuntu package is automatically verified by the packing tool during the installation of the module. The Crypto Officer shall not install the package if the integrity fails.

After the openssl-fips-module-3 package is installed, the Crypto Officer must execute the openssl list -providers command. The Crypto Officer must ensure that the FIPS provider is listed in the output as follows:

fips

name: Ubuntu 24.04 OpenSSL Cryptographic Module

version: 3.0.13-0ubuntu3+Fips1

status: active

The cryptographic boundary consists only of the FIPS provider as listed. If any other OpenSSL or third-party provider is invoked, the user is not interacting with the module specified in this Security Policy.

After, the module needs to be set to run in the FIPS validated configuration. This can be enabled automatically via the Ubuntu Advantage tool after attaching your subscription.

(1) To install the tool type the following commands:

```
$ sudo apt update
$ sudo apt install ubuntu-advantage-tools
```

(2) To activate the Ubuntu Pro subscription run:

```
$ sudo pro attach <your_pro_token>
```

(3) To enable Approved mode run:

```
$ sudo pro enable fips
```

(4) To verify that Approved mode is enabled run:

```
$ sudo pro status
```

The pro client will install the necessary packages for the Approved mode, including the kernel and the bootloader. After this step you MUST **reboot** to put the system into Approved mode. The reboot will boot into FIPS supported kernel and create the /proc/sys/crypto/fips_enabled entry which tells the FIPS certified modules to run in Approved mode. If you do not reboot after installing and configuring the bootloader, Approved mode is not yet enabled.

To verify that FIPS is enabled after the reboot check the /proc/sys/crypto/fips_enabled file and ensure it is set to 1. If it is set to 0, the FIPS modules will not run in Approved mode. If the file is missing, the FIPS kernel is not installed, you can verify that FIPS has been properly enabled with the pro status command.

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 logical interfaces are specified in section 3.1. The requirements and restrictions that shall be considered when operating the module in approved mode are specified in section 2.7 and section 6. The installation, initialization, and startup procedures specified in section 11.1 shall be followed.

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. Then, if desired, the Ubuntu packages can be uninstalled from the Ubuntu 24.04 system.

11.7 Additional Information

Not applicable.

12 Mitigation of Other Attacks

12.1 Attack List

Certain cryptographic subroutines and algorithms are vulnerable to timing analysis. The module mitigates this vulnerability by using constant-time implementations. This includes, but is not limited to:

- Big number operations: computing GCDs, modular inversion, multiplication, division, and modular exponentiation (using Montgomery multiplication).
- Elliptic curve point arithmetic: addition and multiplication (using the Montgomery ladder).
- Vector-based AES implementations.

12.2 Mitigation Effectiveness

RSA, ECDSA, ECDH, and DH employ blinding techniques to further impede timing and power analysis.

12.3 Guidance and Constraints

No configuration is needed to enable the aforementioned countermeasures.

12.4 Additional Information

Not applicable.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
AES-NI	Advanced Encryption Standard New Instructions
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CPACF	CP Assist for Cryptographic Functions
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EVP	Envelope
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
IKE	Internet Key Exchange

KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
KMAC	KECCAK Message Authentication Code
KW	Key Wrap
KWP	Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OAEP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PKCS	Public-Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RSADP	RSA Decryption Primitive
RSAEP	RSA Encryption Primitive
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

ANS X9.42-2001	**Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography** 2001 https://webstore.ansi.org/standards/ascx9/ansix9422001
ANS X9.63-2001	**Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography** 2001 https://webstore.ansi.org/standards/ascx9/ansix9632001
FIPS 140-3	FIPS PUB 140-3 - Security Requirements For Cryptographic Modules March 2019 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf
FIPS 140-3 IG	Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS 180-4	**Secure Hash Standard (SHS)** March 2012 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS 186-5	**Digital Signature Standard (DSS)** February 2023 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf
FIPS 197	**Advanced Encryption Standard** November 2001 https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
FIPS 198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
RFC 3526	**More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)** May 2003 https://www.ietf.org/rfc/rfc3526.txt
RFC 5288	**AES Galois Counter Mode (GCM) Cipher Suites for TLS** August 2008 https://www.ietf.org/rfc/rfc5288.txt
RFC 7919	**Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)** August 2016 https://www.ietf.org/rfc/rfc7919.txt

RFC 8446	**The Transport Layer Security (TLS) Protocol Version 1.3** August 2018 https://www.ietf.org/rfc/rfc8446.txt
SP 800-38A	**Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.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://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf
SP 800-38E	**Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf
SP 800-38F	**Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf
SP 800-52r2	**Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations** August 2019 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf
SP 800-56Ar3	**Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography** April 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf
SP 800-56Cr1	**Recommendation for Key-Derivation Methods in Key-Establishment Schemes** August 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr1.pdf
SP 800-56Cr2	**Recommendation for Key-Derivation Methods in Key-Establishment Schemes** August 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf

SP 800-90Ar1	**Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
SP 800-90B	**Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf
SP 800-108r1	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions** August 2022 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1.pdf
SP 800-132	**Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf
SP 800-133r2	**Recommendation for Cryptographic Key Generation** June 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
SP 800-135r1	**Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf
SP 800-140Br1	**CMVP Security Policy Requirements** November 2023 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Br1.pdf