



Chainguard, Inc.

Chainguard FIPS Provider for OpenSSL

FIPS 140-3 Non-Proprietary Security Policy

Document Version: 1.1

Last Update: 2025-12-27

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

© 2025 Chainguard, Inc., atsec information security.

This document can be reproduced and distributed only whole and intact, including this copyright notice.

Table of Contents

List of Tables.....	4
1 General.....	6
1.1 Overview	6
1.2 Security Levels.....	6
1.3 Additional Information.....	6
2 Cryptographic Module Specification	7
2.1 Description	7
2.2 Tested and Vendor Affirmed Module Version and Identification	7
2.3 Excluded Components	10
2.4 Modes of Operation.....	11
2.5 Algorithms.....	11
2.6 Security Function Implementations.....	16
2.7 Algorithm Specific Information	43
2.7.1 AES XTS	43
2.7.2 Key Derivation using SP 800-132 PBKDF2	43
2.7.3 Compliance to SP 800-56Arev3 Assurances	44
2.7.4 SHA-3	44
2.7.5 Legacy Algorithms.....	44
2.7.6 RSA Signatures.....	44
2.7.7 RSA Key Generation.....	44
2.7.8 Key Transport and Key Agreement	45
2.7.9 AES GCM IV	45
2.7.10 Compliance to SP 800-56Br2 Assurances	45
2.8 RBG and Entropy	45
2.9 Key Generation	46
2.10 Key Establishment.....	47
2.11 Industry Protocols.....	47
3 Cryptographic Module Interfaces.....	48
3.1 Ports and Interfaces.....	48
4 Roles, Services, and Authentication	49
4.1 Authentication Methods.....	49
4.2 Roles.....	49
4.3 Approved Services.....	49
4.4 Non-Approved Services	65

4.5 External Software/Firmware Loaded.....	66
5 Software/Firmware Security	67
5.1 Integrity Techniques	67
5.2 Initiate on Demand	67
6 Operational Environment	68
6.1 Operational Environment Type and Requirements	68
6.2 Configuration Settings and Restrictions.....	68
7 Physical Security	69
8 Non-Invasive Security	70
9 Sensitive Security Parameters Management	71
9.1 Storage Areas	71
9.2 SSP Input-Output Methods	71
9.3 SSP Zeroization Methods.....	71
9.4 SSPs.....	72
9.5 Transitions	84
10 Self-Tests	85
10.1 Pre-Operational Self-Tests.....	85
10.2 Conditional Self-Tests	85
10.3 Periodic Self-Test Information	89
10.4 Error States	91
10.5 Operator Initiation of Self-Tests.....	91
11 Life-Cycle Assurance	92
11.1 Installation, Initialization, and Startup Procedures.....	92
11.2 Administrator Guidance	92
11.3 Non-Administrator Guidance.....	92
11.4 End of Life	92
12 Mitigation of Other Attacks.....	93
12.1 Attack List.....	93
12.2 Mitigation Effectiveness.....	93
12.3 Guidance and Constraints	93
A Glossary and Abbreviations.....	94
B References	96

List of Tables

Table : Security Levels.....	6
Table : Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)	8
Table : Tested Operational Environments - Software, Firmware, Hybrid	8
Table : Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid	10
Table : Modes List and Description	11
Table : Approved Algorithms.....	14
Table : Vendor-Affirmed Algorithms	15
Table : Non-Approved, Not Allowed Algorithms.....	16
Table : Security Function Implementations	43
Table : Entropy Certificates	45
Table : Entropy Sources.....	46
Table : Ports and Interfaces.....	48
Table : Roles.....	49
Table : Approved Services.....	64
Table : Non-Approved Services	66
Table : Storage Areas	71
Table : SSP Input-Output Methods	71
Table : SSP Zeroization Methods.....	72
Table : SSP Table 1	79
Table : SSP Table 2	84
Table : Pre-Operational Self-Tests.....	85
Table : Conditional Self-Tests	89
Table : Pre-Operational Periodic Information.....	89
Table : Conditional Periodic Information	91
Table : Error States	91

List of Figures

Figure 1: Block Diagram.....7

1 General

1.1 Overview

The present document is the non-proprietary FIPS 140-3 Security Policy for the Chainguard FIPS Provider for OpenSSL which will also be referred to as “the module” and “the cryptographic module” throughout this document. This Security Policy specifies the security rules under which the module must operate to meet the FIPS 140-3 Level 1 requirements.

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 Chainguard FIPS Provider for OpenSSL (hereafter referred to as “the module”) is defined as a software module in a multi-chip standalone embodiment. It is a software library that provides a C language application program interface (API) for use by other applications that require cryptographic functionality.

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics:

Cryptographic Boundary: The module consists of one software component, the “FIPS provider” i.e., fips.so (depicted in orange color in Figure 1), that forms the module’s cryptographic boundary which implements the FIPS requirements, and the cryptographic functionality. Components depicted in white color in Figure 1 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 the physical perimeter of the operational environment (a general-purpose computer on which the module is installed) is indicated by a purple dashed line.

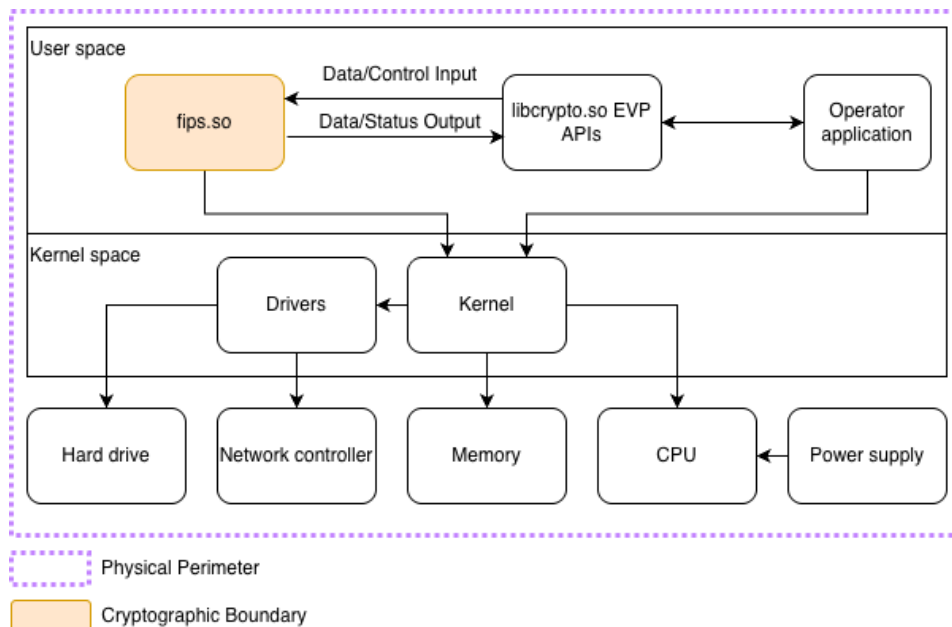


Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Version	Firmware	Features	Integrity Test
fips.so on Chainguard Image 20230214 on Amazon Linux 2023 on EC2 m7i.metal-24xl on Intel Sapphire Rapids Xeon Platinum 8488C	3.4.0-r4		N/A	HMAC-SHA2-256
fips.so on Chainguard Image 20230214 on Amazon Linux 2023 on EC2 m7g.metal on Amazon Graviton3 AWS Graviton3	3.4.0-r4		N/A	HMAC-SHA2-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Chainguard Image 20230214 on Amazon Linux 2023	EC2 m7i.metal-24xl	Intel Sapphire Rapids Xeon Platinum 8488C	No	N/A	3.4.0-r4
Chainguard Image 20230214 on Amazon Linux 2023	EC2 m7g.metal	Amazon Graviton3 AWS Graviton3	No	N/A	3.4.0-r4
Chainguard Image 20230214 on Amazon Linux 2023	EC2 m7i.metal-24xl	Intel Sapphire Rapids Xeon Platinum 8488C	Yes	N/A	3.4.0-r4
Chainguard Image 20230214 on Amazon Linux 2023	EC2 m7g.metal	Amazon Graviton3 AWS Graviton3	Yes	N/A	3.4.0-r4

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

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

Operating System	Hardware Platform
glibc 2.34+ (Host)	Generic Hardware Platform ELF x86-64-v2+
glibc 2.34+ (Host)	Generic Hardware Platform ELF ARMv8a+
AlmaLinux 9	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
AlmaLinux 9	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
AlmaLinux 10	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI

Operating System	Hardware Platform
AlmaLinux 10	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Amazon Linux 2	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Amazon Linux 2	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Amazon Linux 2023	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Amazon Linux 2023	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Azure Linux 2	Azure Esv5-series with Intel Xeon Platinum 8473C, PAA/PAI, under Azure Host Hypervisor
Azure Linux 2	Azure Dpsv6-series with Azure Cobalt 100, PAA/PAI, under Azure Host Hypervisor
Azure Linux 3	Azure Esv5-series with Intel Xeon Platinum 8473C, PAA/PAI, under Azure Host Hypervisor
Azure Linux 3	Azure Dpsv6-series with Azure Cobalt 100, PAA/PAI, under Azure Host Hypervisor
Bottlerocket	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488CC, PAA/PAI
Bottlerocket	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Centos Stream 9	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Centos Stream 9	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Centos Stream 10	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Centos Stream 10	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Chainguard	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Chainguard	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Chainguard (Host)	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Chainguard (Host)	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Chainguard (Host)	Azure Esv5-series with Intel Xeon Platinum 8473C, PAA/PAI, under Azure Host Hypervisor
Chainguard (Host)	Azure Dpsv6-series with Azure Cobalt 100, PAA/PAI, under Azure Host Hypervisor
Chainguard (Host)	GCP c3-highcpu-192-metal with Intel Xeon Platinum 8481C, PAA/PAI, under Titanium
Chainguard (Host)	GCP c4a with Google Axion, PAA/PAI
Chainguard (Host)	Raspberry Pi 5 B Rev 1.0 8GB with Broadcom BCM2712, PAA/PAI
Debian 12 Bookworm	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Debian 12 Bookworm	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI

Operating System	Hardware Platform
Debian 13 Trixie	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Debian 13 Trixie	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Google COS	GCP c3-highcpu-192-metal with Intel Xeon Platinum 8481C
Google COS	GCP c4a with Google Axion, PAA/PAI, under Titanium
RHEL 9	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
RHEL 9	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
RHEL 10	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
RHEL 10	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
RockyLinux 9	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
RockyLinux 9	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
RockyLinux 10	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
RockyLinux 10	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
SUSE Linux Enterprise Server 15 SP6	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
SUSE Linux Enterprise Server 15 SP6	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
SUSE Linux Enterprise Server 16 Public RC	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
SUSE Linux Enterprise Server 16 Public RC	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Ubuntu 20.04	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Ubuntu 20.04	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Ubuntu 22.04	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Ubuntu 22.04	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI
Ubuntu 24.04	AWS EC2 m7i.metal-24x with Intel Xeon Platinum 8488C, PAA/PAI
Ubuntu 24.04	AWS EC2 m7g.metal with AWS Graviton3, PAA/PAI

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

Note: The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated SSPs when the module is ported if the specific operational environments are not listed on the validation certificate.

2.3 Excluded Components

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

2.4 Modes of Operation

Modes List and Description:

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

Table 5: Modes List and Description

After passing all pre-operational self-test and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode of operation.

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 requested service.

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	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CBC-CS1	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CBC-CS2	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CBC-CS3	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CCM	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38C
AES-CFB1	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CFB128	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CFB8	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-CMAC	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38B
AES-CTR	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-ECB	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-GCM	A6662, A6663, A6664, A6665, A6681, A6682, A6692, A6693, A6694, A6695, A6696, A6697, A6698, A6699	-	SP 800-38D
AES-GMAC	A6662, A6663, A6664, A6665, A6681, A6682, A6692, A6693, A6694, A6695, A6696, A6697, A6698, A6699	-	SP 800-38D

Algorithm	CAVP Cert	Properties	Reference
AES-KW	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38F
AES-KWP	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38F
AES-OFB	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38A
AES-XTS Testing Revision 2.0	A6678, A6679, A6680, A6690, A6691, A6700	-	SP 800-38E
Counter DRBG	A6675	-	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 186-5
ECDSA KeyVer (FIPS186-4)	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 186-4
ECDSA KeyVer (FIPS186-5)	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	FIPS 186-5
EDDSA KeyGen	A6676	-	FIPS 186-5
EDDSA KeyVer	A6676	-	FIPS 186-5
EDDSA SigGen	A6676	-	FIPS 186-5
EDDSA SigVer	A6676	-	FIPS 186-5
Hash DRBG	A6675	-	SP 800-90A Rev. 1
HMAC DRBG	A6675	-	SP 800-90A Rev. 1
HMAC-SHA-1	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA2-224	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA2-256	A6666, A6667, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA2-384	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA2-512	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA2- 512/224	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512/256	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 198-1
HMAC-SHA3-224	A6670, A6671, A6687, A6688, A6689	-	FIPS 198-1
HMAC-SHA3-256	A6670, A6671, A6687, A6688, A6689	-	FIPS 198-1
HMAC-SHA3-384	A6670, A6671, A6687, A6688, A6689	-	FIPS 198-1
HMAC-SHA3-512	A6670, A6671, A6687, A6688, A6689	-	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6666, A6668, A6683, A6684, A6685, A6686	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A6674	-	SP 800-56A Rev. 3
KAS-IFC-SSC	A6668	-	SP 800-56A Rev. 3
KDA HKDF SP800-56Cr2	A6673	-	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A6672	-	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A6672	-	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	SP 800-135 Rev. 1
KDF KMAC Sp800-108r1	A6677	-	SP 800-108 Rev. 1
KDF SP800-108	A6677	-	SP 800-108 Rev. 1
KDF SSH (CVL)	A6666, A6668, A6669, A6683, A6684, A6685, A6686	-	SP 800-135 Rev. 1
KMAC-128	A6670, A6671, A6687, A6688, A6689	-	SP 800-185
KMAC-256	A6670, A6671, A6687, A6688, A6689	-	SP 800-185
KTS-IFC	A6666, A6668, A6683, A6684, A6685, A6686	-	SP 800-56B Rev. 2
PBKDF	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	SP 800-132

Algorithm	CAVP Cert	Properties	Reference
RSA KeyGen (FIPS186-5)	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 186-5
RSA SigGen (FIPS186-5)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	FIPS 186-5
RSA SigVer (FIPS186-4)	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 186-4
RSA SigVer (FIPS186-5)	A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689	-	FIPS 186-5
Safe Primes Key Generation	A6674	-	SP 800-56A Rev. 3
Safe Primes Key Verification	A6674	-	SP 800-56A Rev. 3
SHA-1	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-224	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-256	A6666, A6667, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-384	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-512	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-512/224	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA2-512/256	A6666, A6668, A6683, A6684, A6685, A6686	-	FIPS 180-4
SHA3-224	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
SHA3-256	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
SHA3-384	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
SHA3-512	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
SHAKE-128	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
SHAKE-256	A6670, A6671, A6687, A6688, A6689	-	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A6666, A6668, A6683, A6684, A6685, A6686	-	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A6673	-	SP 800-135 Rev. 1

Table 6: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Asymmetric CKG	Key Type: Asymmetric Key pairs: RSA; ECDSA; EdDSA; DH	N/A	SP 800-133r2, section 4, direct DRBG output without XOR

Table 7: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES-GCM encryption with external IV	Encryption
HMAC with key length < 112 bits	Message Authentication Code
KMAC with key length < 112 bits or tag length < 32 bits	Message Authentication Code
TLS 1.2 KDF without extended master secret or with SHA-1 or SHA2-224 or SHA2-512/224 or SHA2-512/256 or SHA-3 functions or with input secret length < 112 bits	Key Derivation Function
KBKDF with input key length < 112 bits	Key Derivation Function
Hash_DRBG or HMAC_DRBG with SHA2-224 or SHA2-384 or SHA2-512/224 or SHA2-512/256 or SHA-3 functions or KMAC	Deterministic Random Bit Generation
ECDH with P-192	Shared Secret Computation
RSA SigVer with modulus length < 2048 bits or PSS salt length > digest length or without hashing (primitive)	Digital Signature Verification
RSA SigGen with SHA-1 or X9.31 padding or modulus length < 2048 bits or PSS salt length > digest length or without hashing (primitive)	Digital Signature Generation
ECDSA SigVer component	Digital Signature Verification
ECDSA SigGen with P-192 or SHA-1; ECDSA SigGen component with P-192	Digital Signature Generation
ECDSA KeyGen with P-192	Key Pair Generation
SSH KDF with SHA2-512/224 or SHA2-512/256 or SHA-3 functions or input secret length < 112 bits	Key Derivation Function
TLS 1.3 KDF with SHA-1 or SHA2-224 or SHA2-512/224 or SHA2-512 or SHA2-512/256 or SHA-3 functions or input secret length < 112 bits	Key Derivation Function
One step KDF, two step KDF, HKDF, ANS X9.42 KDF with input secret length < 112 bits	Key Derivation Function

Name	Use and Function
X448	Provides 224 bits of security, Key Agreement
X25519	Provides 128 bits of security, Key Agreement
ANS X9.63 KDF with SHA-1 or input secret length < 112 bits	Key Derivation Function
RSA OAEP encryption/decryption with 1536-bit modulus	Asymmetric encryption/decryption

Table 8: 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: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS1: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS2: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS3: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB1: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB128: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB8: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CTR: (A6678, A6679, A6680, A6690, A6691, A6700)

Name	Type	Description	Properties	Algorithms
				A6700) AES-ECB: (A6678, A6679, A6680, A6690, A6691, A6700) AES-OFB: (A6678, A6679, A6680, A6690, A6691, A6700) AES-XTS Testing Revision 2.0: (A6678, A6679, A6680, A6690, A6691, A6700)
Decryption with AES	BC-UnAuth	SP 800-38A and SP 800-38E; Decryption		AES-CBC: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS1: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS2: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CBC-CS3: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB1: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB128: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CFB8: (A6678, A6679, A6680, A6690, A6691, A6700) AES-CTR: (A6678, A6679, A6680, A6690, A6691,

Name	Type	Description	Properties	Algorithms
				A6700) AES-ECB: (A6678, A6679, A6680, A6690, A6691, A6700) AES-OFB: (A6678, A6679, A6680, A6690, A6691, A6700) AES-XTS Testing Revision 2.0: (A6678, A6679, A6680, A6690, A6691, A6700)
Authenticated Encryption with AES	BC-Auth	SP 800-38C and SP 800-38D; Authenticated encryption		AES-GCM: (A6662, A6663, A6664, A6665, A6681, A6682, A6692, A6693, A6694, A6695, A6696, A6697, A6698, A6699) AES-CCM: (A6678, A6679, A6680, A6690, A6691, A6700) AES-KW: (A6678, A6679, A6680, A6690, A6691, A6700) AES-KWP: (A6678, A6679, A6680, A6690, A6691, A6700)
Authenticated Decryption with AES	BC-Auth	SP 800-38C and SP 800-38D; Authenticated decryption		AES-GCM: (A6662, A6663, A6664, A6665, A6681, A6682, A6692, A6693, A6694, A6695, A6696, A6697, A6698, A6699) AES-CCM: (A6678, A6679, A6680, A6690, A6691, A6700)

Name	Type	Description	Properties	Algorithms
				A6700) AES-KW: (A6678, A6679, A6680, A6690, A6691, A6700) AES-KWP: (A6678, A6679, A6680, A6690, A6691, A6700)
Message Authentication Code (MAC) Generation with AES	MAC	SP 800-38B and SP 800-38D; MAC Generation		AES-GMAC: (A6662, A6663, A6664, A6665, A6681, A6682, A6692, A6693, A6694, A6695, A6696, A6697, A6698, A6699) AES-CMAC: (A6678, A6679, A6680, A6690, A6691, A6700)
Message Authentication Code (MAC) Generation with HMAC	MAC	FIPS 198-1; MAC Generation		HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512/224: (A6666, A6668, A6683,

Name	Type	Description	Properties	Algorithms
				A6684, A6685, A6686) HMAC-SHA2- 512/256: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA3-224: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684,

Name	Type	Description	Properties	Algorithms
				A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Message Authentication Code (MAC) Generation with KMAC	MAC	SP 800-185; MAC Generation		KMAC-128: (A6670, A6671, A6687, A6688, A6689) KMAC-256: (A6670, A6671, A6687, A6688, A6689)
Random Number Generation with a DRBG	DRBG	SP 800-90Arev1; Random Number Generation		Counter DRBG: (A6675) Hash DRBG: (A6675) HMAC DRBG: (A6675) AES-ECB: (A6678, A6679, A6680, A6690, A6691, A6700) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683,

Name	Type	Description	Properties	Algorithms
				A6684, A6685, A6686) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686)
Signature Generation with RSA	DigSig-SigGen	FIPS 186-5; Signature Generation		RSA SigGen (FIPS186-5): (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668,

Name	Type	Description	Properties	Algorithms
				A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Signature Verification with RSA	DigSig-SigVer	FIPS 186-5; FIPS 186-4; Signature Verification		RSA SigVer (FIPS186-4): (A6666, A6668, A6683, A6684, A6685, A6686) RSA SigVer (FIPS186-5): (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683,

Name	Type	Description	Properties	Algorithms
				A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Signature Generation with ECDSA	DigSig-SigGen	FIPS 186-5; Signature Generation		ECDSA SigGen (FIPS186-5): (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685,

Name	Type	Description	Properties	Algorithms
				A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Signature Verification with ECDSA	DigSig-SigVer	FIPS 186-5; FIPS 186-4; Signature Verification		ECDSA SigVer (FIPS186-4): (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) ECDSA SigVer (FIPS186-5): (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686)

Name	Type	Description	Properties	Algorithms
				SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Signature Generation with EdDSA	DigSig-SigGen	FIPS 186-5; Signature Generation		EDDSA SigGen: (A6676) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHAKE-256: (A6670, A6671, A6687, A6688, A6689)
Signature Verification with EdDSA	DigSig-SigVer	FIPS 186-5; Signature Verification		EDDSA SigVer: (A6676) SHA2-512: (A6666,

Name	Type	Description	Properties	Algorithms
				A6668, A6683, A6684, A6685, A6686) SHAKE-256: (A6670, A6671, A6687, A6688, A6689)
Key Generation ECDSA	Pair with	AsymKeyPair- KeyGen	FIPS 186-5; Key Pair Generation	ECDSA KeyGen (FIPS186-5): (A6666, A6668, A6683, A6684, A6685, A6686)
Key Generation RSA	Pair with	AsymKeyPair- KeyGen	FIPS 186-5; Key Pair Generation	RSA KeyGen (FIPS186-5): (A6666, A6668, A6683, A6684, A6685, A6686)
Key Generation EdDSA	Pair with	AsymKeyPair- KeyGen	FIPS 186-5; Key Pair Generation	EDDSA KeyGen: (A6676) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHAKE-256: (A6670, A6671, A6687, A6688, A6689)
Key Generation Safe Primes	Pair with	AsymKeyPair- KeyGen	SP 800-56Arev3; Key Pair Generation	Safe Primes Key Generation: (A6674)
Key Verification Safe Primes	Pair with	AsymKeyPair- KeyVer	SP 800-56Arev3; Key Pair Verification	Safe Primes Key Verification: (A6674)
Public Verification ECDSA	Key with	AsymKeyPair- KeyVer	FIPS 186-5; Key Pair Verification	ECDSA KeyVer (FIPS186-4): (A6666, A6668, A6683, A6684, A6685, A6686) ECDSA KeyVer (FIPS186-5): (A6666, A6668, A6683, A6684, A6685, A6686)

Name	Type	Description	Properties	Algorithms
Public Key Verification with EdDSA	AsymKeyPair- KeyVer	FIPS 186-5; Key Pair Verification		EDDSA KeyVer: (A6676)
Key Derivation with KBKDF	KBKDF	SP 800-108rev1; Key Derivation		KDF SP800-108: (A6677) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/256: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA3-224: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384:

Name	Type	Description	Properties	Algorithms
				(A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687,

Name	Type	Description	Properties	Algorithms
				A6688, A6689) KDF KMAC Sp800-108r1: (A6677) KMAC-128: (A6670, A6671, A6687, A6688, A6689) KMAC-256: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with KDA OneStep	KAS-56CKDF	SP 800-56Crev2; Key Derivation		KDA OneStep SP800-56Cr2: (A6672) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA3-224: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687,

Name	Type	Description	Properties	Algorithms
				A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689) KMAC-128: (A6670, A6671, A6687, A6688, A6689) KMAC-256: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with KDA TwoStep	KAS-56CKDF	SP 800-56Crev2; Key Derivation		KDA TwoStep SP800-56Cr2: (A6672) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/224: (A6666, A6668, A6683,

Name	Type	Description	Properties	Algorithms
				A6684, A6685, A6686) HMAC-SHA2- 512/256: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA3-224: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684,

Name	Type	Description	Properties	Algorithms
				A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with KDA HKDF	KAS-56CKDF	SP 800-56Crev2; Key Derivation		KDA HKDF SP800- 56Cr2: (A6673) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/224: (A6666, A6668, A6683, A6684, A6685, A6686)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA3-224: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256:

Name	Type	Description	Properties	Algorithms
				(A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with ANS X9.42 KDF	KAS-135KDF	SP 800-135rev1; Key Derivation		KDF ANS 9.42: (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684,

Name	Type	Description	Properties	Algorithms
				A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with ANS X9.63 KDF	KAS-135KDF	SP 800-135rev1; Key Derivation		KDF ANS 9.63: (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256:

Name	Type	Description	Properties	Algorithms
				(A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Key Derivation with SSH KDF	KAS-135KDF	SP 800-135rev1; Key Derivation		KDF SSH: (A6666, A6668, A6669, A6683, A6684, A6685, A6686) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686)
Key Derivation with TLS 1.2 KDF	KAS-135KDF	SP 800-135rev1; Key Derivation		TLS v1.2 KDF RFC7627: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666,

Name	Type	Description	Properties	Algorithms
				A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686)
Key Derivation with TLS 1.3 KDF	KAS-135KDF	RFC 8446; Key Derivation		TLS v1.3 KDF: (A6673) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686)
Key Derivation with PBKDF2	PBKDF	SP 800-132; Key Derivation		PBKDF: (A6666, A6668, A6670, A6671, A6683, A6684, A6685, A6686, A6687, A6688, A6689) HMAC-SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2-512:

Name	Type	Description	Properties	Algorithms
				(A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/224: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA2- 512/256: (A6666, A6668, A6683, A6684, A6685, A6686) HMAC-SHA3-224: (A6671, A6687, A6688, A6689) HMAC-SHA3-256: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-384: (A6670, A6671, A6687, A6688, A6689) HMAC-SHA3-512: (A6670, A6671, A6687, A6688, A6689) SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683,

Name	Type	Description	Properties	Algorithms
				A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Shared Secret Computation	KAS-SSC	SP 800-56Ar3 KAS-ECC-SSC with P-224, P-256, P-384, and P-521 (these respectively support strengths of 112, 128, 192, and 256 bits); SP 800-56Ar3 KAS-FFC-SSC with MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192, ffdhe-2048, ffdhe-3072, ffdhe-4096, ffdhe-6144, ffdhe-8192 (these respectively support strengths of 112, 128, 152, 176, 200, 112, 128, 152, 176, and 200 bits);		KAS-FFC-SSC Sp800-56Ar3: (A6674) KAS-ECC-SSC Sp800-56Ar3: (A6666, A6668, A6683, A6684, A6685, A6686) KAS-IFC-SSC: (A6668)

Name	Type	Description	Properties	Algorithms
		SP 800-56Br2 KAS-IFC-SSC		
Message Digest with SHA	SHA	FIPS 180-4 and FIPS 202; Message Digest		SHA-1: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-256: (A6666, A6667, A6668, A6683, A6684, A6685, A6686) SHA2-384: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/224: (A6666, A6668, A6683, A6684, A6685, A6686) SHA2-512/256: (A6666, A6668, A6683, A6684, A6685, A6686) SHA3-224: (A6670, A6671, A6687, A6688, A6689) SHA3-256: (A6670, A6671, A6687, A6688, A6689) SHA3-384: (A6670, A6671, A6687, A6688, A6689) SHA3-512: (A6670, A6671, A6687, A6688, A6689)
Message Digest with SHAKE	XOF	FIPS 202; Message Digest		SHAKE-128: (A6670, A6671, A6687, A6688,

Name	Type	Description	Properties	Algorithms
				A6689) SHAKE-256: (A6670, A6671, A6687, A6688, A6689)
Key encapsulation	KTS-Encap	SP 800-56Br2; RSA key encapsulation		KTS-IFC: (A6666, A6668, A6683, A6684, A6685, A6686)
Key un- encapsulation	KTS-Encap	SP 800-56Br2; RSA key un- encapsulation		KTS-IFC: (A6666, A6668, A6683, A6684, A6685, A6686)

Table 9: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES XTS

In accordance with 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 into 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, Section 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.

2.7.2 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 with 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 supports 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 supported 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 (shorter salts are not supported by the module), shall be generated randomly using the SP 800-90Ar1 DRBG provided by the module.

- The iteration count shall be selected as large as possible, if the time required to generate the key using the entered password is acceptable for the users. The minimum iteration count supported by the module is 1000.

2.7.3 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). 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.

2.7.4 SHA-3

The module implements the SHA-3 functions as both standalone functions and as part of higher-level algorithms (in compliance with FIPS 140-3 IG C.C). As detailed in Section 2.6 Security Function Implementations with corresponding certificates, the cryptographic algorithms that use SHA-3 functions include RSA signature generation and verification, ECDSA signature generation and verification, EdDSA signature generation and verification, EdDSA key pair generation, KBKDF, KDA HKDF, X9.63 KDF, X9.42 KDF, PBKDF, OneStep KDA, TwoStep KDA, and HMAC. In addition, the implementation of the extendable output functions SHAKE128 and SHAKE256 were verified to have a standalone usage.

2.7.5 Legacy Algorithms

The module utilizes the following legacy algorithms as defined in SP 800-131Arev2:

- SHA-1 for RSA Signature Verification and ECDSA Signature Verification purposes.
- ECDSA Signature Verification, under FIPS 186-4, allows verifying elliptic curve-based signatures with curve P-192.

2.7.6 RSA Signatures

The module's RSA signature generation and verification implementations were CAVP tested with moduli sizes 2048, 3072, and 4096 bits. These moduli sizes are allowed by FIPS 140-3 IG C.F. The module supports moduli sizes larger than 4096 bits.

2.7.7 RSA Key Generation

The module's RSA key generation implementation was CAVP tested with moduli sizes 2048, 3072, 4096, 6144, and 8192 bits. These moduli sizes are allowed by FIPS 140-3 IG C.F. The module supports moduli sizes larger than 8192 bits. The number of Miller-Rabin tests is compliant with Table B.1 of FIPS 186-5.

2.7.8 Key Transport and Key Agreement

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.

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.

2.7.9 AES GCM IV

For TLS 1.2 and 1.3, the module offers the AES GCM implementation per Scenario 1 and 5 of the FIPS 140-3 IG C.H respectively. For TLS 1.2 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. For TLS 1.3 defined in RFC8446, the module supports AES GCM cipher suites from Section 3.3.1 of SP800-52r2.

The module does not implement the TLS 1.2 or 1.3 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 module also implements GCM with internal IV generation per Scenario 2 of IG C.H. The internally generated IVs are always 96 bits and are generated using the approved DRBG internal to the module's boundary.

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.

2.7.10 Compliance to SP 800-56Br2 Assurances

To comply with the assurances found in Section 6.4 of SP 800-56Br2, the operator must use the module in the context of the TLS or SSH protocols. Additionally, the module's approved key pair generation service (see Section 4.3) must be used to generate RSA key pairs, or the key pairs must be obtained from another FIPS-validated module. As part of this service, the module will internally perform the key pair validation of the generated public key.

The operator must use the `EVP_PKEY_public_check()` API to perform partial public key validation of the peer public key, complying with Section 6.4.2.2 of SP 800-56Br2. The operator must also confirm the peer's possession of private key by using any method specified in Section 6.4.2.3 of SP 800-56Br2.

2.8 RBG and Entropy

Cert Number	Vendor Name
E191	Chainguard, Inc

Table 10: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Chainguard CPU Time Jitter RNG Entropy Source	Non-Physical	Chainguard Image 20230214 on Amazon Linux 2023 on EC2 m7g.metal on Amazon Graviton3 AWS Graviton3; Chainguard Image 20230214 on Amazon Linux 2023 on EC2 m7i.metal-24xl on Intel Sapphire Rapids Xeon Platinum 8488C	256	full entropy	SHA3-256 (A5446)

Table 11: Entropy Sources

The entropy source is statically compiled into the module and hence it is located within the cryptographic boundary of the module. As per the Public document of entropy certificate E191, the entropy source provides full entropy of 256 bits.

In addition to the DRBG algorithms provided to the operator, the module internally uses two dedicated DRBG instances based on SP 800-90A Rev. 1 to generate seeds for asymmetric key pairs and random numbers for security functions. The following parameters are used:

1. Private DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate secret random values (e.g. during asymmetric key pair generation). It can be accessed using `RAND_priv_bytes`.
2. Public DRBG: AES-256 CTR_DRBG with derivation function. This DRBG is used to generate general purpose random values that do not need to remain secret (e.g. initialization vectors). It can be accessed using `RAND_bytes`.

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133r2. When random values are required, they are obtained from the SP 800-90Ar1 approved DRBG of type CTR_DRBG, compliant with Section 4 of SP 800-133r2. This method does not use the value V as described in Additional Comment 2 of FIPS 140-3 IG D.H. The following methods are implemented:

- Safe primes key pair generation: Compliant with SP 800-133r2, Section 5.2, which maps to SP 800-56Ar3.
- RSA key pair generation: Compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-5.
- ECC (ECDH and ECDSA) key pair generation: Compliant with SP 800-133r2, Section 5.1 and Section 5.2, which maps to FIPS 186-5.
- EdDSA key pair generation: Compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-5.

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

Additionally, the module implements the following key derivation methods, per SP 800-133r2, Section 6.2:

- KBKDF: compliant with SP 800-108r1. This implementation can be used to derive secret keys from a pre-existing key-derivation-key.
- KDA OneStep, KDA TwoStep, HKDF: compliant with SP 800-56Cr2. These implementations shall only be used to derive secret keys in the context of an SP 800-56Ar3 key agreement scheme.

- ANS X9.42 KDF (CVL), ANS X9.63 KDF (CVL): compliant with SP 800-135r1. These implementations shall only be used to derive secret keys in the context of an ANS X9.42-2001 resp. ANS X9.63- 2001 key agreement scheme.
- SSH KDF (CVL), TLS 1.2 KDF (CVL), TLS 1.3 KDF (CVL): compliant with SP 800-135r1 and RFC 8446. These implementations shall only be used to derive secret keys in the context of the SSH, TLS 1.2, or TLS 1.3 protocols, respectively.
- PBKDF2: compliant with option 1a of SP 800-132. This implementation shall only be used to derive keys for use in storage applications.

2.10 Key Establishment

The module implements key establishment methods as listed in the Security Function Implementations table in Section 2.6.

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 (except for 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.

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 12: Ports and Interfaces

As a software-only module, the module does not have physical ports. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The module does not implement any authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 13: Roles

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

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Message Digest	Compute a message digest	EVP_DigestFinal returns 1	Message	Message digest	Message Digest with SHA Message Digest with SHAKE	Crypto Officer
Symmetric Encryption	Encrypt a plaintext	EVP_EncryptFinal_ex returns 1	AES Key, plaintext, IV	Ciphertext	Encryption with AES	Crypto Officer - AES Key: W,E
Symmetric Decryption	Decrypt a ciphertext	EVP_DecryptFinal_ex returns 1	AES Key, ciphertext, IV	Plaintext	Decryption with AES	Crypto Officer - AES Key: W,E
Authenticated Symmetric Encryption	Encrypt and authenticate a plaintext	GCM: OSSL_CIPHER_PARAM_AEAD_IV_GENERATED is 1; CCM: EVP_EncryptFinal_ex returns 1	AES Key, plaintext, IV	Ciphertext, MAC tag	Authenticated Encryption with AES	Crypto Officer - AES Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Authenticated Symmetric Decryption	Decrypt and authenticate a ciphertext	EVP_DecryptFinal_ex returns non-negative value	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	EVP_MAC_final returns 1	AES Key, message	MAC tag	Message Authentication Code (MAC) Generation with AES	Crypto Officer - AES Key: W,E
HMAC Message Authentication Generation	Compute a MAC tag using HMAC	OSSL_MAC_PARAM_FIPS_APPROVED_INDICATOR is 1	HMAC Key, message	MAC tag	Message Authentication Code (MAC) Generation with HMAC	Crypto Officer - HMAC Key: W,E
KMAC Message Authentication Generation	Compute a MAC tag using KMAC	OSSL_MAC_PARAM_FIPS_APPROVED_INDICATOR is 1	KMAC Key, message	MAC tag	Message Authentication Code (MAC) Generation with KMAC	Crypto Officer - KMAC Key: W,E
TLS KDF Key Derivation	TLS key derivation	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
KBKDF Key Derivation	Derive a key from a key-derivation key	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	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	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	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	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	Shared Secret	ANS X9.63 Derived Key	Key Derivation with ANS 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	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	Shared Secret	HKDF Derived key	Key Derivation with KDA HKDF	Crypto Officer - Shared Secret: W,E - HKDF Derived Key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
OneStep KDA Key Derivation	Derive a key from a shared secret	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	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	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	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	OSSL_KDF_PARAM_FIPS_APPROVED_INDICATOR is 1	Shared Secret	SSH KDF Derived Key	Key Derivation with SSH KDF	Crypto Officer - Shared Secret: W,E - SSH KDF Derived Key: G,R
PBKDF Key Derivation	Derive a key from a password	EVP_KDF_derive returns 1	Password	PBKDF Derived Key	Key Derivation with PBKDF2	Crypto Officer - Password: W,E - PBKDF Derived Key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Random Number Generation	Generate random number	OSSL_RAND_PARAM_FIPS_APPROVED_INDICATOR is 1	Number of bits	Random number	Random Number Generation with a DRBG	Crypto Officer - Entropy Input: W,E - DRBG Seed: G,E - DRBG Internal State (V, Key): G,W,E - DRBG Internal State (V, C): G,W,E
Shared Secret Computation	Compute a shared secret	KAS-FFC-SSC: EVP_PKEY_derive returns 1; KAS-ECC-SSC: OSSL_EXCHANGE_PARAM_FIPS_APPROVED_INDICATOR is 1	DH Private Key (owner), DH Public Key (peer); EC Private Key (owner), EC Public Key (peer); RSA Private Key (owner), RSA Public	Shared Secret	Shared Secret Computation	Crypto Officer - Shared Secret: G,R - DH Private Key: W,E - DH Public Key: W,E - EC Private Key: W,E - EC Public Key: W,E - RSA Private

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			Key (peer)			Key: W,E - RSA Public Key: W,E
RSA Digital Signature Generation	Generate a digital signature with RSA	OSSL_SIGNATURE_PARAM_FIPS_APPROVED_INDICATOR is 1 and OSSL_SIGNATURE_PARAM_FIPS_VERIFY_MESSAGE is 1	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	OSSL_SIGNATURE_PARAM_FIPS_APPROVED_INDICATOR is 1 and OSSL_SIGNATURE_PARAM_FIPS_VERIFY_MESSAGE is 1	EC Private Key, message, hash algorithm	Signature	Signature Generation with ECDSA	Crypto Officer - EC Private Key: W,E
EdDSA Digital Signature Generation	Generate a digital signature with EdDSA	EVP_PKEY_sign returns 1	EdDSA Private Key, message, hash algorithm	Signature	Signature Generation with EdDSA	Crypto Officer - EdDSA Private Key: W,E
RSA Digital Signature Verification	Verify a digital signature using RSA	OSSL_SIGNATURE_PARAM_FIPS_APPROVED_INDICATOR is 1 and OSSL_SIGNATURE_PARAM_FIPS_VERIFY_MESSAGE is 1	RSA Public Key, message, signature, hash algorithm	Pass or Fail	Signature Verification with RSA	Crypto Officer - RSA Public Key: W,E
ECDSA Digital Signature	Verify a digital signature	OSSL_SIGNATURE_PARAM_FIPS_APPROVED_INDICATOR is 1 and OSSL_SIGNATURE_PARAM_FIPS_VERIFY_MESSAGE is 1	EC Public Key, message	Pass or Fail	Signature Verification with ECDSA	Crypto Officer - EC Public

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Verification	Verify using ECDSA		Signature, hash algorithm			Key: W,E
EdDSA Digital Signature Verification	Verify a digital signature using EdDSA	EVP_PKEY_verify returns 1	EdDSA Public Key, message, signature, hash algorithm	Pass or Fail	Signature Verification with EdDSA	Crypto Officer - EdDSA Public Key: W,E
RSA Key Pair Generation	Generate an RSA key pair	EVP_PKEY_keygen returns 1	Module Length	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	Generate	OSSL_PKEY_PARAM_FIPS_APPROVED_INDICATOR is 1	Curve	Module Generated	Key Pair Generation	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Generation	EC key pair			ed EC Private Key, Module Generated EC Public Key	on with ECDSA	- Module Generated EC Private Key: G,R - Module Generated EC Public Key: G,R - Intermediate Key Generation Value: G,E,Z
Eddsa Key Pair Generation	Generate an Eddsa key pair	EVP_PKEY_keygen returns 1	Curve	Module Generated Eddsa Private Key, Module Generated Eddsa Public Key	Key Pair Generation with Eddsa	Crypto Officer - Module Generated Eddsa Private Key: G,R - Module Generated Eddsa Public Key: G,R - Intermediate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key Generation Value: G,E,Z
Key Pair Generation with Safe Primes	Generate an DH key pair	EVP_PKEY_keygen returns 1	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
Public Key Verification with ECDSA	Verify an EC key pair	EVP_PKEY_public_check returns 1	EC Private Key, EC Public Key	Pass or Fail	Public Key Verification with ECDSA	Crypto Officer - EC Private Key: W,E - EC Public Key: W,E
Key Pair Verification with	Verify a DH	EVP_PKEY_check or EVP_PKEY_public_check or EVP_PKEY_private_check returns 1	DH Private Key,	Pass or Fail	Key Pair Verification with	Crypto Officer - DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Safe Primes	key pair		DH Public Key		Safe Primes	Private Key: W,E - DH Public Key: W,E
Public Key Verification with EdDSA	Verify an EdDSA key pair	EVP_PKEY_public_check returns 1	EdDSA Public Key, EdDSA Private Key	Pass or Fail	Public Key Verification with EdDSA	Crypto Officer - EdDSA Private Key: W,E - EdDSA Public Key: W,E
Show Version	Return the name and version information	None	None	Module name and version	None	Crypto Officer
Show Status	Return the module status	None	None	Module status	None	Crypto Officer
Self-Test	Perform the CASTs and integrity test	None	None	Pass or Fail of self-tests	Decryption with AES Authenticated Encryption with AES Authenticated Decryption with	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					AES Random Number Generation with a DRBG Signature Generation with RSA Signature Verification with RSA Signature Generation with ECDSA Signature Verification with ECDSA Signature Generation with EdDSA Signature Verification with EdDSA Key Derivation with KBKDF Key Derivation with KDA OneStep Key Derivation with KDA HKDF	

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Key Derivation with ANS X9.42 KDF Key Derivation with ANS X9.63 KDF Key Derivation with TLS 1.2 KDF Key Derivation with TLS 1.3 KDF Key Derivation with PBKDF2 Shared Secret Computation Message Digest with SHA	
Zeroization	Zeroize any SSP	None	An SSP	None	None	Crypto Officer - AES Key: Z - HMAC Key: Z - KMAC

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						KDF Derived Key: Z - Entropy Input: Z - DRBG Internal State (V, Key): Z - DRBG Seed: Z - DH Private Key: Z - DH Public Key: Z - EC Private Key: Z - EC Public Key: Z - EdDSA Public Key: Z - EdDSA Private Key: Z - RSA Private Key: Z - RSA Public Key: Z - Module Generated DH Private

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

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Private Key: Z - Module Generated EdDSA Public Key: Z - DRBG Internal State (V, C): Z
Key encapsulation	Encapsulate a key	OSSL_ASYM_CIPHER_PARAM_FIPS_A PPROVED_INDICATOR is 1	RSA Public Key, Key to encapsulate	Encapsulated key	Key encapsulation	Crypto Officer - RSA Public Key: W,E
Key un-encapsulation	Un-encapsulate a key	OSSL_ASYM_CIPHER_PARAM_FIPS_A PPROVED_INDICATOR is 1	RSA Private Key, Key to un-encapsulate	Un-encapsulated key	Key un-encapsulation	Crypto Officer - RSA Private Key: W,E

Table 14: 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 defines the services that utilize 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.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES-GCM with Externally Generated IV	Encrypt a plaintext	AES-GCM encryption with external IV	Crypto Officer
Message Authentication Code Generation with Non-Approved Key Length or Tag Length	Compute a MAC tag	HMAC with key length < 112 bits KMAC with key length < 112 bits or tag length < 32 bits	Crypto Officer
Key Derivation with Non-Approved Parameters	Derive a key	TLS 1.2 KDF without extended master secret or with SHA-1 or SHA2-224 or SHA2-512/224 or SHA2-512/256 or SHA-3 functions or with input secret length < 112 bits KBKDF with input key length < 112 bits SSH KDF with SHA2-512/224 or SHA2-512/256 or SHA-3 functions or input secret length < 112 bits TLS 1.3 KDF with SHA-1 or SHA2-224 or SHA2-512/224 or SHA2-512 or SHA2-512/256 or SHA-3 functions or input secret length < 112 bits One step KDF, two step KDF, HKDF, ANS X9.42 KDF with input secret length < 112 bits ANS X9.63 KDF with SHA-1 or input secret length < 112 bits	Crypto Officer
Random Number Generation	Generate a random number	Hash_DRBG or HMAC_DRBG with SHA2-224 or SHA2-384 or SHA2-512/224 or SHA2-512/256 or SHA-3 functions or KMAC	Crypto Officer
Shared Secret Computation with Non-Approved Parameters	Compute a shared secret	ECDH with P-192	Crypto Officer
Non-approved Digital Signature Generation	Generate a digital signature on a pre-hashed message	RSA SigGen with SHA-1 or X9.31 padding or modulus length < 2048 bits or PSS salt length > digest	Crypto Officer

Name	Description	Algorithms	Role
		length or without hashing (primitive) ECDSA SigGen with P-192 or SHA-1; ECDSA SigGen component with P-192	
Non-approved Digital Signature Verification	Verify a digital signature of a pre-hashed message	RSA SigVer with modulus length < 2048 bits or PSS salt length > digest length or without hashing (primitive) ECDSA SigVer component	Crypto Officer
ECDSA Key Pair Generation with Non-Approved Parameters	Generate an EC key pair	ECDSA KeyGen with P-192	Crypto Officer
Key Exchange	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module)	X448 X25519	Crypto Officer
Asymmetric Encryption/Decryption	Perform RSA OAEP encryption/decryption	RSA OAEP encryption/decryption with 1536-bit modulus	Crypto Officer

Table 15: Non-Approved Services

4.5 External Software/Firmware Loaded

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

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC-SHA2-256 value calculated at runtime 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 tests can be invoked on demand by unloading and subsequently re-initializing the module (i.e., rebooting the system), which will perform (among others) the software integrity tests.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

Any SSPs contained within the module are protected by the process isolation and memory separation mechanisms, and only the module has control over these SSPs.

If the operating system is properly installed, it 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.

7 Physical Security

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

8 Non-Invasive Security

The module does not implement any non-invasive security mechanisms.

9 Sensitive Security Parameters Management

9.1 Storage Areas

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

Table 16: Storage Areas

The module does not perform persistent storage of SSPs; SSPs in use by the module exist in volatile memory only. SSPs are provided to the module by the calling process and are destroyed when released by the appropriate zeroization function calls.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Input Parameters	Operator calling application (TOEPP)	Cryptographic Module	Plaintext	Manual	Electronic	
API Output Parameters	Cryptographic Module	Operator Calling Application (TOEPP)	Plaintext	Manual	Electronic	

Table 17: SSP Input-Output Methods

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.

9.3 SSP Zeroization Methods

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

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. The successful completion of the running service indicates that zeroization was completed.	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. The successful completion of the module reset indicates that the zeroization procedure succeeded.	By unloading and reloading the module

Table 18: SSP Zeroization Methods

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES Key	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 Code (MAC) Generation with AES

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
HMAC Key	Used for hash-based message authentication	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Code (MAC) Generation with HMAC
KMAC Key	Used for message authentication	128-1024 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Code (MAC) Generation with KMAC
Key-Derivation Key	Used for key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key Derivation with KBKDF
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 ANS 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

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
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 ANS 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		
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		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Entropy Input	Used for random number generation and seeding a DRBG (compliant with IG D.L)	128-384 bits - 128-384 bits of entropy	Entropy input - CSP			Random Number Generation with a DRBG
DRBG Internal State (V, Key)	Used for random number generation (compliant with IG D.L)	Counter DRBG V length: 128 bits; Counter DRBG Key length: 128, 192, 256 bits; HMAC DRBG V length: 160, 256, 512 bits; HMAC DRBG Key length: 160, 256, 512 bits - Counter DRBG: 128, 192, 256 bits; HMAC DRBG: 128, 256 bits	Internal state - CSP	Random Number Generation with a DRBG		Random Number Generation with a DRBG
DRBG Internal State (V, C)	Used for random number generation (compliant with IG D.L)	440, 888 bits - 128, 256 bits	Internal state - CSP	Random Number Generation with a DRBG		Random Number Generation with a DRBG
DRBG Seed	Used for random number generation (compliant with IG D.L)	192-888 bits - 128-256 bits	Seed - CSP	Random Number Generation with a DRBG		Random Number Generation with a DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DH Private Key	Used for shared secret computation and key pair verification	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112-200 bits	Private key - CSP			Key Pair Verification with Safe Primes Shared Secret Computation
DH Public Key	Used for shared secret computation and key pair verification	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 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 - 112-256 bits	Private key - CSP			Signature Generation with ECDSA Public Key Verification with ECDSA Shared Secret Computation
EC Public Key	Used for shared secret computation,	P-192, P-224, P-256, P-384, P-	Public key - PSP			Signature Verification with ECDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	signature verification, and key pair verification	521 - 96-256 bits				Public Key Verification with ECDSA Shared Secret Computation
EdDSA Private Key	Used for digital signature generation, and key pair verification	Ed25519, Ed448 - 128, 224 bits	Private key - CSP			Signature Generation with EdDSA Public Key Verification with EdDSA
EdDSA Public Key	Used for signature verification, and key pair verification	Ed25519, Ed448 - 128, 224 bits	Public key - PSP			Signature Verification with EdDSA Public Key Verification with EdDSA
RSA Private Key	Used for signature generation, shared secret computation, and key un-encapsulation	2048-16384 bits - 112-256 bits	Private key - CSP			Signature Generation with RSA Shared Secret Computation Key un-encapsulation
RSA Public Key	Used for signature verification, shared secret computation, and key encapsulation	2048-16384 bits - 112-256 bits	Public key - PSP			Signature Verification with RSA Shared Secret Computation Key encapsulation
Module Generated DH Private Key	DH private key generated by the module	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096,	Private key - CSP	Key Pair Generation with Safe Primes		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		MODP-6144, MODP-8192 - 112-200 bits				
Module Generated DH Public Key	DH public key generated by the module	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 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 - 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 - 112-256 bits	Public key - PSP	Key Pair Generation with ECDSA		
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		Key Pair Generation with ECDSA Key Pair Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
				Generation with RSA Key Pair Generation with EdDSA Key Pair Generation with Safe Primes		with RSA Key Pair Generation with EdDSA Key Pair Generation with Safe Primes
Module Generated EdDSA Private Key	EdDSA private key generated by the module	Ed25519, - Ed448 - 128, 224 bits	Private key - CSP	Key Pair Generation with EdDSA		
Module Generated EdDSA Public Key	EdDSA public key generated by the module	Ed25519, - Ed448 - 128, 224 bits	Public key - CSP	Key Pair Generation with EdDSA		

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	
HMAC Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	
KMAC Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	
Key-Derivation Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	KBKDF Derived Key:Derives
Shared Secret	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle	DH Private Key:Generated From DH Public

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API Output Parameters			Module Reset	Key:Generated From EC Private Key:Generated From EC Public Key:Generated From
Password	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	PBKDF Derived Key:Derives
PBKDF Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Password:Derived From
KBKDF Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Key-Derivation Key:Derived From
ANS X9.42 Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
ANS X9.63 Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
HKDF Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
KDA OneStep Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
KDA TwoStep Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
TLS Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle	Shared Secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Module Reset	
SSH KDF Derived Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Shared Secret:Derived From
Entropy Input		RAM:Plaintext	Until the DRBG has completed instantiation or module is reset	Automatic Module Reset	DRBG Seed:Generates
DRBG Internal State (V, Key)		RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	DRBG Seed:Generated From
DRBG Internal State (V, C)		RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	DRBG Seed:Generated From
DRBG Seed		RAM:Plaintext	Until the DRBG has completed instantiation or module is reset	Automatic 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	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	DH Public Key:Paired With Shared Secret:Derives
DH Public Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	DH Private Key:Paired With Shared Secret:Derives
EC Private Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	EC Public Key:Paired With Shared Secret:Derives
EC Public Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle	EC Private Key:Paired With Shared Secret:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
				Module Reset	
EdDSA Private Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	EdDSA Public Key:Paired With
EdDSA Public Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	EdDSA Private Key:Paired With
RSA Private Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	RSA Public Key:Paired With Shared Secret:Derives
RSA Public Key	API Input Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	RSA Private Key:Paired With Shared Secret:Derives
Module Generated DH Private Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	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	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Module Generated DH Private Key:Paired With Intermediate Key Generation Value:Generated From
Module Generated EC Private Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Module Generated EC Public Key:Paired With Intermediate Key Generation Value:Generated From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Module Generated EC Public Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	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	Until cipher handle is freed or module is reset	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	Until cipher handle is freed or module is reset	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, or until module is reset	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 Module Generated EdDSA Private Key:Generates Module Generated

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					EdDSA Public Key:Generates
Module Generated EdDSA Private Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Module Generated EdDSA Public Key:Paired With Intermediate Key Generation Value:Generated from
Module Generated EdDSA Public Key	API Output Parameters	RAM:Plaintext	Until cipher handle is freed or module is reset	Free Cipher Handle Module Reset	Module Generated EdDSA Private Key:Paired With Intermediate Key Generation Value:Generated from

Table 20: SSP Table 2

9.5 Transitions

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

10 Self-Tests

While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module does not return control to the calling application until the tests are completed. If any of these tests fail, the module transitions to the error state.

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256	256-bit key	Message Authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test for fips.so

Table 21: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is powered on, before the module transitions to the operational state.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB	Decrypt with 128-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	Encrypt and 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
SHA2-512	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	4-byte message	KAT	CAST	Module becomes operational and services	Message digest	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				are available for use		
Counter DRBG	128 bit keys, DF, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1 including health test per section 11.3	Test runs at power-on before the integrity test
HMAC DRBG	HMAC-SHA-1, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1 including health test per section 11.3	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3	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	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	HMAC-SHA2-256 with 128-bit key; KMAC-128 with 128-bit key	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	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
KDF ANS 9.42	SHA-1	KAT	CAST	Module becomes operational	Industry-based ANS X9.42 key derivation	Test runs at power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				and services are available for use		before the integrity test
KDF ANS 9.63	SHA2-256	KAT	CAST	Module becomes operational and services are available for use	Industry-based ANS X9.63 key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627	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	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
PBKDF	HMAC-SHA2-256 with 200-bit derived key length, 24-character password, 4096 iterations, 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	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)	P-224 with 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
ECDSA SigVer (FIPS186-5)	P-224 with 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 SigGen (FIPS186-5)	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
EDDSA SigGen (FIPS186-5)	Ed25519; Ed4488	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)	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
EDDSA SigVer (FIPS186-5)	Ed25519; Ed4488	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)	SHA2-512	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
EDDSA KeyGen (FIPS186-5)	N/A	PCT	PCT	Successful key pair generation	Signature generation & verification	Key pair generation
RSA KeyGen (FIPS186-5)	Section 6.4.1.1 of SP800-56Br2	PCT	PCT	Successful key pair generation	Encryption & decryption	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Safe Primes Key Generation	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
Hash DRBG	SHA2-256, with PR	KAT	CAST	Module becomes operational and services are available for use	Compliant with SP 800-90Ar1 including health test per section 11.3	Test runs at power-on before the integrity test
Entropy Source - RCT and APT start-up test	1024 samples. Repetition count test according to Section 4.4.1 and Adaptive proportion test according to Section 4.4.2 of SP 800-90B	RCT and APT	CAST	Module becomes operational and services are available for use	Entropy source start-up test	Entropy source initialization
Entropy Source - RCT and APT continuous test	Repetition count test according to Section 4.4.1 and Adaptive proportion test according to Section 4.4.2 of SP 800-90B	RCT and APT	CAST	Entropy source is operational	Entropy source continuous test	Continuously when the entropy source is accessed

Table 22: Conditional Self-Tests

10.3 Periodic Self-Test Information

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

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB	KAT	CAST	On Demand	Manually
AES-GCM	KAT	CAST	On Demand	Manually
SHA2-512	KAT	CAST	On Demand	Manually
SHA3-256	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG	KAT	CAST	On Demand	Manually
HMAC DRBG	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3	KAT	CAST	On Demand	Manually
KDF SP800-108	KAT	CAST	On Demand	Manually
KDA OneStep SP800-56Cr2	KAT	CAST	On Demand	Manually
KDF ANS 9.42	KAT	CAST	On Demand	Manually
KDF ANS 9.63	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627	KAT	CAST	On Demand	Manually
TLS v1.3 KDF	KAT	CAST	On Demand	Manually
PBKDF	KAT	CAST	On Demand	Manually
KDA HKDF SP800- 56Cr2	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5)	KAT	CAST	On Demand	Manually
EDDSA SigGen (FIPS186-5)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5)	KAT	CAST	On Demand	Manually
EDDSA SigVer (FIPS186-5)	KAT	CAST	On Demand	Manually
ECDSA KeyGen (FIPS186-5)	PCT	PCT	On Demand	Manually
EDDSA KeyGen (FIPS186-5)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-5)	PCT	PCT	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Safe Primes Key Generation	PCT	PCT	On Demand	Manually
Hash DRBG	KAT	CAST	On Demand	Manually
Entropy Source - RCT and APT start-up test	RCT and APT	CAST	On Demand	Manually
Entropy Source - RCT and APT continuous test	RCT and APT	CAST	On Demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Power-up error	An error occurred during the integrity test or CAST failure	Software integrity test failure CAST failure	Re-initialization of the module	Module not loaded
PCT error	An error occurred during a PCT	PCT failure	Re-initialization of the module	Cryptographic functionality is blocked

Table 25: Error States

In any error state, the output interface is inhibited, and the module cannot perform cryptographic operations.

10.5 Operator Initiation of Self-Tests

The software integrity test and cryptographic algorithm self-tests can be invoked on demand by unloading and subsequently re-initializing the module. The PCTs can be invoked on demand by requesting the key pair generation service.

11 Life-Cycle Assurance

All configuration items are uniquely identified by a compound value consisting of the package apk full version and revision number plus architecture with the git commit SHA1 value.

11.1 Installation, Initialization, and Startup Procedures

No installation, initialization, or startup steps are required as the module is pre-built into the images that the vendor provides.

11.2 Administrator Guidance

To verify that the module is prebuilt into a Chainguard image, the Crypto Officer must review the image specification (e.g. Chainguard Console SBOM tab, SPDX image attestation, apk installed package database, syft output, AWS Inspector) to ensure it contains the package “openssl-fips-provider-3.4.0” version “3.4.0-r4”.

The Crypto Officer must verify the name and version of the module. This is done by retrieving the parameters `OSSL_PROV_PARAM_NAME` and `OSSL_PROV_PARAM_BUILDINFO` from the module. Since there can be several providers for OpenSSL, and since only the “fips” provider is the module, the Crypto Officer must ensure that the “fips” provider of OpenSSL is queried when retrieving these parameters and when requesting any other services.

`OSSL_PROV_PARAM_NAME` must have the value: “Chainguard FIPS Provider for OpenSSL”

`OSSL_PROV_PARAM_BUILDINFO` must have the value: “3.4.0-r4”

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.

11.3 Non-Administrator Guidance

There is no non-administrator guidance.

11.4 End of Life

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

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).
- EdDSA implementations
- 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.

A Glossary and Abbreviations

AES	Advanced Encryption Standard
AESNI	Advanced Encryption Standard New Instructions
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM	Counter with Cipher Block Chain-Message Authentication Code
CCP	Cryptographic Co-Processor
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
CTS	Ciphertext Stealing
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptographic
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
HMAC	Keyed-Hash Message Authentication Code
IG	International Guidance
IKE	Internet Key Exchange
IV	Initialization Vector

KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-Based 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
OEAP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PBKDF2	Password-based Key Derivation Function v2
PCT	Pair-wise Consistency Test
PKI	Public Key Infrastructure
PSP	Public Security Parameter
PSS	Probabilistic Signature Scheme
RSA	Rivest Shamir Adleman
RSADP	RSA Decryption Primitive
RSAEP	RSA Encryption Primitive
SHA	Secure Hash Algorithm
SHAKE	Secure Hash Algorithm with Keccak
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
TOEPP	Tested Operational Environment's Physical Perimeter
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

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 IG** **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**
<https://csrc.nist.gov/CSRC/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf>
- FIPS 180-4** **Secure Hash Standard (SHS)**
August 2015
<https://doi.org/10.6028/NIST.FIPS.180-4>
- FIPS 186-4** **Digital Signature Standard (DSS)**
July 2013
<https://doi.org/10.6028/NIST.FIPS.180-4>
- FIPS 186-5** **Digital Signature Standard (DSS)**
February 2023
<https://doi.org/10.6028/NIST.FIPS.186-5>
- FIPS 197** **Advanced Encryption Standard (AES)**
November 2001; Updated May 2023
<https://doi.org/10.6028/NIST.FIPS.197-upd1>
- FIPS 198-1** **The Keyed-Hash Message Authentication Code (HMAC)**
July 2008
<https://doi.org/10.6028/NIST.FIPS.198-1>
- FIPS 202** **SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions**
August 2015
<https://doi.org/10.6028/NIST.FIPS.202>
- PKCS#1** **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 800-90Ar1	Recommendation for Random Number Generation Using Deterministic Random Bit Generators June 2015 https://doi.org/10.6028/NIST.SP.800-90Ar1
SP 800-90B	Recommendation for the Entropy Sources Used for Random Bit Generation January 2018 https://doi.org/10.6028/NIST.SP.800-90B
SP 800-108r1	Recommendation for Key Derivation Using Pseudorandom Functions August 2022; Updated February 2024 https://doi.org/10.6028/NIST.SP.800-108r1-upd1
SP 800-131Ar2	Transitioning the Use of Cryptographic Algorithms and Key Lengths March 2019 https://doi.org/10.6028/NIST.SP.800-131Ar2
SP 800-133r2	Recommendation for Cryptographic Key Generation June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP 800-140Br1	Cryptographic Module Validation Program (CMVP) Security Policy Requirements: CMVP Validation Authority Updates to ISO/IEC 24759 and ISO/IEC 19790 Annex B November 2023 https://doi.org/10.6028/NIST.SP.800-140Br1