Cloudlinux Inc., TuxCare division

OpenSSL FIPS Provider for AlmaLinux 9

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.1
Last update: 2024-09-27

Prepared by:

atsec information security corporation

4516 Seton Center Pkwy, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for version 3.0.7-1d2bd88ee26b3c90 of the OpenSSL FIPS Provider for AlmaLinux 9. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 1 module.

This Non-Proprietary Security Policy may be reproduced and distributed, but only whole and intact and including this notice. Other documentation is proprietary to their authors.

1.2 Security Levels

Section	Security Level
1	1
2	1
3	1
4	1
5	1
6	1
7	N/A
8	N/A
9	1
10	1
11	1
12	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The OpenSSL FIPS Provider for AlmaLinux 9 (hereafter referred to as “the module”) is defined as a software module in a multi-chip standalone embodiment. It provides a C language application program interface (API) for use by other applications that require cryptographic functionality. The module is a software library supporting FIPS 140-3 approved algorithms developed by TuxCare for its use by other applications that require cryptographic functionality and consists of one software component, the “FIPS provider”, which implements the FIPS requirements and the cryptographic functionality provided to the operator.

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics:

Cryptographic Boundary:

The cryptographic boundary of the module is defined as the fips.so shared library, which contains the compiled code implementing the FIPS provider.

Tested Operational Environment’s Physical Perimeter (TOEPP):

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

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

Green lines indicate the flow of data between the cryptographic module and its operator application, through the logical interfaces defined in Section 3 Cryptographic Module Interfaces.

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

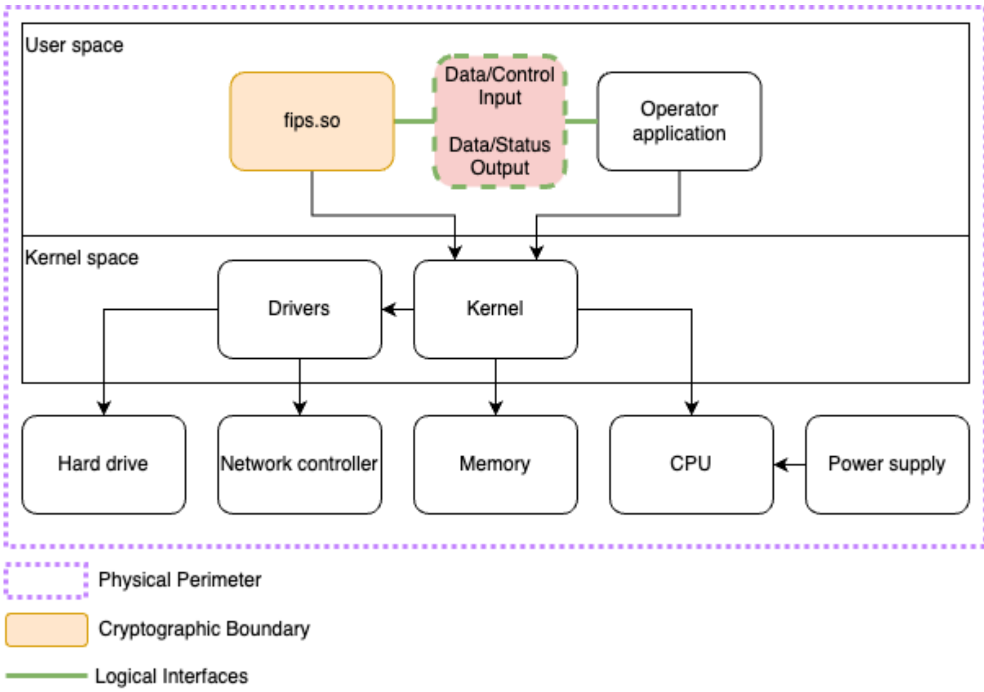

Figure 1 - Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.so	3.0.7-1d2bd88ee26b3c90	N/A	HMAC-SHA2-256

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

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
AlmaLinux 9.2	Amazon Web Services	Intel Xeon Platinum 8259CL	With and without PAA (AES-NI and	N/A	3.0.7-1d2bd88ee26b3c90

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
	(AWS) m5.metal		SHA Extensions)		
AlmaLinux 9.2	Amazon Web Services (AWS) a1.metal	AWS Graviton	With and without PAA (Neon and Crypto Extensions)	N/A	3.0.7-1d2bd88ee26b3c90

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

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

N/A for this module.

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

2.3 Excluded Components

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

2.4 Modes of Operation

Modes List and Description:

Table 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 4: Modes List and Description

After passing all pre-operational self-tests and cryptographic algorithm self-tests executed on start-up, the module automatically transitions to the approved mode. No operator intervention is required to reach this point. The module operates in the approved mode of operation by default and can only transition into the non-approved mode by calling one of the non-approved services listed in the Non-Approved Services table of the Security Policy.

In the operational state, the module accepts service requests from calling applications through its logical interfaces. At any point in the operational state, a calling application can end its process, causing the module to end its 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 service that was requested.

Degraded Mode Description:

The module does not implement a degraded mode of operation.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Reference
KDA OneStep SP800-56Cr2	A4051	SP 800-56C Rev. 2
KDA HKDF Sp800-56Cr1	A4052	SP 800-56C Rev. 2
TLS v1.3 KDF (CVL)	A4052	SP 800-135 Rev. 1
Counter DRBG	A4053	SP 800-90A Rev. 1
Hash DRBG	A4053	SP 800-90A Rev. 1
HMAC DRBG	A4053	SP 800-90A Rev. 1
AES-ECB	A4054	SP 800-38A
KDF SSH (CVL)	A4054	SP 800-135 Rev. 1
ECDSA SigGen (FIPS186-4)	A4055	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4055	FIPS 186-4
HMAC-SHA3-224	A4055	FIPS 198-1
HMAC-SHA3-256	A4055	FIPS 198-1
HMAC-SHA3-384	A4055	FIPS 198-1
HMAC-SHA3-512	A4055	FIPS 198-1
KDF ANS 9.42 (CVL)	A4055	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4055	SP 800-135 Rev. 1
PBKDF	A4055	SP 800-132
SHA3-224	A4055	FIPS 202
SHA3-256	A4055	FIPS 202

Algorithm	CAVP Cert	Reference
SHA3-384	A4055	FIPS 202
SHA3-512	A4055	FIPS 202
SHAKE-128	A4055	FIPS 202
SHAKE-256	A4055	FIPS 202
AES-CBC	A4056	SP 800-38A
AES-CBC-CS1	A4056	SP 800-38A
AES-CBC-CS2	A4056	SP 800-38A
AES-CBC-CS3	A4056	SP 800-38A
AES-CCM	A4056	SP 800-38C
AES-CFB1	A4056	SP 800-38A
AES-CFB128	A4056	SP 800-38A
AES-CFB8	A4056	SP 800-38A
AES-CMAC	A4056	SP 800-38B
AES-CTR	A4056	SP 800-38A
AES-ECB	A4056	SP 800-38A
AES-KW	A4056	SP 800-38F
AES-KWP	A4056	SP 800-38F
AES-OFB	A4056	SP 800-38A
AES-XTS Testing Revision 2.0	A4056	SP 800-38E
AES-CBC	A4057	SP 800-38A
AES-CBC-CS1	A4057	SP 800-38A
AES-CBC-CS2	A4057	SP 800-38A
AES-CBC-CS3	A4057	SP 800-38A
AES-CCM	A4057	SP 800-38C

Algorithm	CAVP Cert	Reference
AES-CFB1	A4057	SP 800-38A
AES-CFB128	A4057	SP 800-38A
AES-CFB8	A4057	SP 800-38A
AES-CMAC	A4057	SP 800-38B
AES-CTR	A4057	SP 800-38A
AES-ECB	A4057	SP 800-38A
AES-KW	A4057	SP 800-38F
AES-KWP	A4057	SP 800-38F
AES-OFB	A4057	SP 800-38A
AES-XTS Testing Revision 2.0	A4057	SP 800-38E
AES-CBC	A4058	SP 800-38A
AES-CBC-CS1	A4058	SP 800-38A
AES-CBC-CS2	A4058	SP 800-38A
AES-CBC-CS3	A4058	SP 800-38A
AES-CCM	A4058	SP 800-38C
AES-CFB1	A4058	SP 800-38A
AES-CFB128	A4058	SP 800-38A
AES-CFB8	A4058	SP 800-38A
AES-CMAC	A4058	SP 800-38B
AES-CTR	A4058	SP 800-38A
AES-ECB	A4058	SP 800-38A
AES-KW	A4058	SP 800-38F
AES-KWP	A4058	SP 800-38F
AES-OFB	A4058	SP 800-38A

Algorithm	CAVP Cert	Reference
AES-XTS Testing Revision 2.0	A4058	SP 800-38E
ECDSA KeyGen (FIPS186-4)	A4059	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4059	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4059	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4059	FIPS 186-4
HMAC-SHA-1	A4059	FIPS 198-1
HMAC-SHA2-224	A4059	FIPS 198-1
HMAC-SHA2-256	A4059	FIPS 198-1
HMAC-SHA2-384	A4059	FIPS 198-1
HMAC-SHA2-512	A4059	FIPS 198-1
HMAC-SHA2-512/224	A4059	FIPS 198-1
HMAC-SHA2-512/256	A4059	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4059	SP 800-56A Rev. 3
KDF ANS 9.42 (CVL)	A4059	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4059	SP 800-135 Rev. 1
PBKDF	A4059	SP 800-132
RSA KeyGen (FIPS186-4)	A4059	FIPS 186-4
RSA SigGen (FIPS186-4)	A4059	FIPS 186-4
RSA SigVer (FIPS186-4)	A4059	FIPS 186-4
SHA-1	A4059	FIPS 180-4
SHA2-224	A4059	FIPS 180-4
SHA2-256	A4059	FIPS 180-4
SHA2-384	A4059	FIPS 180-4
SHA2-512	A4059	FIPS 180-4

Algorithm	CAVP Cert	Reference
SHA2-512/224	A4059	FIPS 180-4
SHA2-512/256	A4059	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4059	SP 800-135 Rev. 1
HMAC-SHA2-256	A4060	FIPS 198-1
SHA2-256	A4060	FIPS 180-4
AES-CBC	A4061	SP 800-38A
AES-CBC-CS1	A4061	SP 800-38A
AES-CBC-CS2	A4061	SP 800-38A
AES-CBC-CS3	A4061	SP 800-38A
AES-CCM	A4061	SP 800-38C
AES-CFB1	A4061	SP 800-38A
AES-CFB128	A4061	SP 800-38A
AES-CFB8	A4061	SP 800-38A
AES-CMAC	A4061	SP 800-38B
AES-CTR	A4061	SP 800-38A
AES-ECB	A4061	SP 800-38A
AES-KW	A4061	SP 800-38F
AES-KWP	A4061	SP 800-38F
AES-OFB	A4061	SP 800-38A
AES-XTS Testing Revision 2.0	A4061	SP 800-38E
AES-CBC	A4062	SP 800-38A
AES-CBC-CS1	A4062	SP 800-38A
AES-CBC-CS2	A4062	SP 800-38A
AES-CBC-CS3	A4062	SP 800-38A

Algorithm	CAVP Cert	Reference
AES-CCM	A4062	SP 800-38C
AES-CFB1	A4062	SP 800-38A
AES-CFB128	A4062	SP 800-38A
AES-CFB8	A4062	SP 800-38A
AES-CMAC	A4062	SP 800-38B
AES-CTR	A4062	SP 800-38A
AES-ECB	A4062	SP 800-38A
AES-KW	A4062	SP 800-38F
AES-KWP	A4062	SP 800-38F
AES-OFB	A4062	SP 800-38A
AES-XTS Testing Revision 2.0	A4062	SP 800-38E
AES-CBC	A4063	SP 800-38A
AES-CBC-CS1	A4063	SP 800-38A
AES-CBC-CS2	A4063	SP 800-38A
AES-CBC-CS3	A4063	SP 800-38A
AES-CCM	A4063	SP 800-38C
AES-CFB1	A4063	SP 800-38A
AES-CFB128	A4063	SP 800-38A
AES-CFB8	A4063	SP 800-38A
AES-CMAC	A4063	SP 800-38B
AES-CTR	A4063	SP 800-38A
AES-ECB	A4063	SP 800-38A
AES-KW	A4063	SP 800-38F
AES-KWP	A4063	SP 800-38F

Algorithm	CAVP Cert	Reference
AES-OFB	A4063	SP 800-38A
AES-XTS Testing Revision 2.0	A4063	SP 800-38E
AES-ECB	A4064	SP 800-38A
KDF SSH (CVL)	A4064	SP 800-135 Rev. 1
AES-ECB	A4065	SP 800-38A
KDF SSH (CVL)	A4065	SP 800-135 Rev. 1
AES-ECB	A4066	SP 800-38A
KDF SSH (CVL)	A4066	SP 800-135 Rev. 1
AES-GCM	A4067	SP 800-38D
AES-GMAC	A4067	SP 800-38D
AES-GCM	A4068	SP 800-38D
AES-GMAC	A4068	SP 800-38D
AES-GCM	A4069	SP 800-38D
AES-GMAC	A4069	SP 800-38D
AES-GCM	A4070	SP 800-38D
AES-GMAC	A4070	SP 800-38D
AES-GCM	A4071	SP 800-38D
AES-GMAC	A4071	SP 800-38D
AES-GCM	A4072	SP 800-38D
AES-GMAC	A4072	SP 800-38D
AES-GCM	A4073	SP 800-38D
AES-GMAC	A4073	SP 800-38D
AES-GCM	A4074	SP 800-38D
AES-GMAC	A4074	SP 800-38D

Algorithm	CAVP Cert	Reference
AES-GCM	A4075	SP 800-38D
AES-GMAC	A4075	SP 800-38D
AES-GCM	A4076	SP 800-38D
AES-GMAC	A4076	SP 800-38D
AES-GCM	A4077	SP 800-38D
AES-GMAC	A4077	SP 800-38D
AES-GCM	A4078	SP 800-38D
AES-GMAC	A4078	SP 800-38D
ECDSA KeyGen (FIPS186-4)	A4079	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4079	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4079	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4079	FIPS 186-4
HMAC-SHA-1	A4079	FIPS 198-1
HMAC-SHA2-224	A4079	FIPS 198-1
HMAC-SHA2-256	A4079	FIPS 198-1
HMAC-SHA2-384	A4079	FIPS 198-1
HMAC-SHA2-512	A4079	FIPS 198-1
HMAC-SHA2-512/224	A4079	FIPS 198-1
HMAC-SHA2-512/256	A4079	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4079	SP 800-56A Rev. 3
KDF ANS 9.42 (CVL)	A4079	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4079	SP 800-135 Rev. 1
PBKDF	A4079	SP 800-132
RSA KeyGen (FIPS186-4)	A4079	FIPS 186-4

Algorithm	CAVP Cert	Reference
RSA SigGen (FIPS186-4)	A4079	FIPS 186-4
RSA SigVer (FIPS186-4)	A4079	FIPS 186-4
SHA-1	A4079	FIPS 180-4
SHA2-224	A4079	FIPS 180-4
SHA2-256	A4079	FIPS 180-4
SHA2-384	A4079	FIPS 180-4
SHA2-512	A4079	FIPS 180-4
SHA2-512/224	A4079	FIPS 180-4
SHA2-512/256	A4079	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4079	SP 800-135 Rev. 1
ECDSA KeyGen (FIPS186-4)	A4080	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4080	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4080	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4080	FIPS 186-4
HMAC-SHA-1	A4080	FIPS 198-1
HMAC-SHA2-224	A4080	FIPS 198-1
HMAC-SHA2-256	A4080	FIPS 198-1
HMAC-SHA2-384	A4080	FIPS 198-1
HMAC-SHA2-512	A4080	FIPS 198-1
HMAC-SHA2-512/224	A4080	FIPS 198-1
HMAC-SHA2-512/256	A4080	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4080	SP 800-56A Rev. 3
KDF ANS 9.42 (CVL)	A4080	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4080	SP 800-135 Rev. 1

Algorithm	CAVP Cert	Reference
PBKDF	A4080	SP 800-132
RSA KeyGen (FIPS186-4)	A4080	FIPS 186-4
RSA SigGen (FIPS186-4)	A4080	FIPS 186-4
RSA SigVer (FIPS186-4)	A4080	FIPS 186-4
SHA-1	A4080	FIPS 180-4
SHA2-224	A4080	FIPS 180-4
SHA2-256	A4080	FIPS 180-4
SHA2-384	A4080	FIPS 180-4
SHA2-512	A4080	FIPS 180-4
SHA2-512/224	A4080	FIPS 180-4
SHA2-512/256	A4080	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4080	SP 800-135 Rev. 1
ECDSA KeyGen (FIPS186-4)	A4081	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4081	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4081	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4081	FIPS 186-4
HMAC-SHA-1	A4081	FIPS 198-1
HMAC-SHA2-224	A4081	FIPS 198-1
HMAC-SHA2-256	A4081	FIPS 198-1
HMAC-SHA2-384	A4081	FIPS 198-1
HMAC-SHA2-512	A4081	FIPS 198-1
HMAC-SHA2-512/224	A4081	FIPS 198-1
HMAC-SHA2-512/256	A4081	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4081	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Reference
KDF ANS 9.42 (CVL)	A4081	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4081	SP 800-135 Rev. 1
PBKDF	A4081	SP 800-132
RSA KeyGen (FIPS186-4)	A4081	FIPS 186-4
RSA SigGen (FIPS186-4)	A4081	FIPS 186-4
RSA SigVer (FIPS186-4)	A4081	FIPS 186-4
SHA-1	A4081	FIPS 180-4
SHA2-224	A4081	FIPS 180-4
SHA2-256	A4081	FIPS 180-4
SHA2-384	A4081	FIPS 180-4
SHA2-512	A4081	FIPS 180-4
SHA2-512/224	A4081	FIPS 180-4
SHA2-512/256	A4081	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4081	SP 800-135 Rev. 1
ECDSA KeyGen (FIPS186-4)	A4082	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4082	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A4082	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4082	FIPS 186-4
HMAC-SHA-1	A4082	FIPS 198-1
HMAC-SHA2-224	A4082	FIPS 198-1
HMAC-SHA2-256	A4082	FIPS 198-1
HMAC-SHA2-384	A4082	FIPS 198-1
HMAC-SHA2-512	A4082	FIPS 198-1
HMAC-SHA2-512/224	A4082	FIPS 198-1

Algorithm	CAVP Cert	Reference
HMAC-SHA2-512/256	A4082	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A4082	SP 800-56A Rev. 3
KDF ANS 9.42 (CVL)	A4082	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4082	SP 800-135 Rev. 1
PBKDF	A4082	SP 800-132
RSA KeyGen (FIPS186-4)	A4082	FIPS 186-4
RSA SigGen (FIPS186-4)	A4082	FIPS 186-4
RSA SigVer (FIPS186-4)	A4082	FIPS 186-4
SHA-1	A4082	FIPS 180-4
SHA2-224	A4082	FIPS 180-4
SHA2-256	A4082	FIPS 180-4
SHA2-384	A4082	FIPS 180-4
SHA2-512	A4082	FIPS 180-4
SHA2-512/224	A4082	FIPS 180-4
SHA2-512/256	A4082	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A4082	SP 800-135 Rev. 1
KDF SP800-108	A4084	SP 800-108 Rev. 1
KAS-FFC-SSC Sp800-56Ar3	A4085	SP 800-56A Rev. 3
Safe Primes Key Generation	A4085	SP 800-56A Rev. 3
Safe Primes Key Verification	A4085	SP 800-56A Rev. 3

Table 5: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
(CKG) Key Pair Generation	Key Pair Generation with RSA:2048-16384 bits keys (112-256 bits strength) Key Pair Generation with ECDSA:P-224,	N/A	SP 800-133r2 Section 4 example 1

Name	Properties	Implementation	Reference
	P-256, P-384, P-521 curves (112, 128, 192, 256 bits strength) Key Pair Generation with Safe Primes:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 groups (112-200 bits) Key Type:Asymmetric		
RSA Signature Generation	PKCS#1v1.5 and PKCSPSS:SHA3-224, SHA3-256, SHA3-384, SHA3-512 Key:2048-16384 bits Strength:112-256 bits	N/A	FIPS 140-3 IG C.C
RSA Signature Verification	PKCS#1v1.5 and PKCSPSS:SHA3-224, SHA3-256, SHA3-384, SHA3-512 Key:1024-16384 bits Strength:80-256 bits	N/A	FIPS 140-3 IG C.C

Table 6: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

The module does not implement non-approved algorithms that are allowed in the approved mode of operation with no security claimed.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
AES GCM (external IV)	Authentication Encryption
HMAC (< 112-bit keys)	Message Authentication Code
KBKDF, KDA OneStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys)	Key Derivation
KDA OneStep (SHAKE128, SHAKE256)	Key Derivation
ANS X9.42 KDF (SHAKE128, SHAKE256)	Key Derivation
ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256)	Key Derivation

Name	Use and Function
SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256)	Key Derivation
TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3)	TLS Key Derivation
TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	TLS Key Derivation
PBKDF2 (< 8 characters password; < 128 salt length; < 1000 iterations; < 112-bit keys)	Password-based Key Derivation
RSA (KAS1, KAS2 schemes)	Shared secret computation
RSA and ECDSA (pre-hashed message)	Signature generation; Signature verification
RSA-PSS (invalid salt length)	Signature generation; Signature verification
RSA-OAEP	Asymmetric encryption; Asymmetric decryption

Table 7: Non-Approved, Not Allowed Algorithms

The table above lists all non-approved cryptographic algorithms of the module employed by the non-approved services of the Non-Approved Services table in Section 4.4 Non-Approved Services.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric Encryption with AES	BC-UnAuth	Symmetric encryption using AES	AES-ECB:128, 192, 256 bits AES-CBC:128, 192, 256 bits AES-CBC-CS1:128, 192, 256 bits AES-CBC-CS2:128, 192, 256 bits AES-CBC-CS3:128, 192, 256 bits AES-CFB1:128, 192, 256 bits AES-CFB128:128, 192, 256 bits AES-CFB8:128, 192, 256 bits AES-CTR:128, 192, 256	AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC-

Name	Type	Description	Properties	Algorithms
			bits AES-OFB:128, 192, 256 bits AES-XTS Testing Revision 2.0:128, 256 bits	CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8

Name	Type	Description	Properties	Algorithms
				AES-CFB8 AES-CFB8 AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Symmetric Decryption with AES	BC-UnAuth	Symmetric decryption using AES	AES-ECB:128, 192, 256 bits AES-CBC:128, 192, 256 bits AES-CBC-CS1:128, 192, 256 bits AES-CBC-CS2:128, 192, 256 bits AES-CBC-CS3:128, 192, 256 bits AES-CFB1:128, 192, 256 bits AES-CFB128:128, 192, 256 bits AES-CFB8:128, 192, 256 bits AES-CTR:128, 192, 256	AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC-

Name	Type	Description	Properties	Algorithms
			bits AES-OFB:128, 192, 256 bits AES-XTS Testing Revision 2.0:128, 256 bits	CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS1 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS2 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CBC-CS3 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB1 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8

Name	Type	Description	Properties	Algorithms
				AES-CFB8 AES-CFB8 AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-OFB AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0 AES-XTS Testing Revision 2.0
Key wrapping with AES-KW	KTS-Wrap	Key wrapping using AES KW	AES-KW:128, 192, 256 bits	AES-KW AES-KW AES-KW AES-KW AES-KW AES-KW
Key unwrapping with AES-KW	KTS-Wrap	Key unwrapping using AES KW	AES-KW:128, 192, 256 bits	AES-KW AES-KW AES-KW AES-KW AES-KW AES-KW
Key wrapping with AES-KWP	KTS-Wrap	Key wrapping using AES KW with padding	AES-KWP:128, 192, 256 bits	AES-KWP AES-KWP AES-KWP

Name	Type	Description	Properties	Algorithms
				AES-KWP AES-KWP AES-KWP
Key unwrapping with AES-KWP	KTS-Wrap	Key unwrapping using AES KW with padding	AES-KWP:128, 192, 256 bits	AES-KWP AES-KWP AES-KWP AES-KWP AES-KWP AES-KWP
Key Derivation with KDA OneStep	KAS-56CKDF	Key Derivation using KDA OneStep	KDA OneStep SP800-56Cr2:112-256 bits	KDA OneStep SP800-56Cr2
Key Derivation with X9.42 KDF	KAS-135KDF	Key Derivation using X9.42 KDF	KDF ANS 9.42:112-256 bits	KDF ANS 9.42 KDF ANS 9.42 KDF ANS 9.42 KDF ANS 9.42 KDF ANS 9.42 KDF ANS 9.42
Key Derivation with X9.63 KDF	KAS-135KDF	Key Derivation using X9.63 KDF	KDF ANS 9.63:112-256 bits	KDF ANS 9.63 KDF ANS 9.63 KDF ANS 9.63 KDF ANS 9.63 KDF ANS 9.63 KDF ANS 9.63
Key Derivation with SSH KDF	KAS-135KDF	Key Derivation	KDF SSH:112-256 bits	KDF SSH KDF SSH KDF SSH KDF SSH
Key Derivation with HKDF	KAS-56CKDF	Key Derivation using HKDF	KDA HKDF Sp800-56Cr1:112-256 bits	KDA HKDF Sp800-56Cr1

Name	Type	Description	Properties	Algorithms
TLS Key Derivation	KAS-135KDF	TLS 1.2 / 1.3 Key Derivation	TLS v1.3 KDF:112-256 bits TLS v1.2 KDF RFC7627:112-256 bits	TLS v1.3 KDF TLS v1.2 KDF RFC7627 TLS v1.2 KDF RFC7627 TLS v1.2 KDF RFC7627 TLS v1.2 KDF RFC7627 TLS v1.2 KDF RFC7627
Key Derivation with KBKDF	KBKDF	Key derivation using KBKDF	Modes:Counter, Feedback KDF SP800-108:112-256 bits	KDF SP800-108
Password-based Key Derivation	PBKDF	Password-based Key Derivation	PBKDF:112-256 bits	PBKDF PBKDF PBKDF PBKDF PBKDF PBKDF
Random Number Generation	DRBG	Random Number Generation (IG D.R compliant)	Counter DRBG:128, 192, 256 bits HMAC DRBG:128, 256 bits Hash DRBG:128, 256 bits	Counter DRBG HMAC DRBG Hash DRBG
Signature Generation	DigSig-SigGen	Signature Generation	ECDSA SigGen (FIPS186-4):P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits) RSA SigGen (FIPS186-4):2048-16384 bits (112-256 bits)	ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4)

Name	Type	Description	Properties	Algorithms
				ECDSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4)
Signature Verification	DigSig-SigVer	Signature Verification	ECDSA SigVer (FIPS186-4):P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits) RSA SigVer (FIPS186-4):NIST SP 800-131Ar2 Legacy use: 1024-2047 bits with 80-111 bits of security strength; NIST SP 800-131Ar2 Acceptable: 2048-16384 bits with 112-256 bits of security strength IG C.F Compliance:The module supports RSA modulus sizes which are not tested by CAVP in compliance with FIPS 140-3 IG C.F	ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4)
Message Authentication Code	MAC	Message Authentication Code	HMAC hashes:SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 AES key:128, 192, 256 bits	HMAC-SHA3-224 HMAC-SHA3-256 HMAC-SHA3-384 HMAC-SHA3-512

Name	Type	Description	Properties	Algorithms
				HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512/224 HMAC-SHA2-512/224

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/224 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 AES-CMAC AES-CMAC AES-CMAC AES-CMAC AES-CMAC AES-CMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC AES-GMAC
Message digest	SHA XOF	Message digest		SHA3-224 SHA3-256 SHA3-384 SHA3-512 SHAKE-128 SHAKE-256 SHA-1 SHA-1 SHA-1 SHA-1 SHA-1 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-224

Name	Type	Description	Properties	Algorithms
				SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/224 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256
Authenticated Symmetric Encryption	BC-Auth	Authenticated Symmetric Encryption	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM

Name	Type	Description	Properties	Algorithms
				AES-GCM AES-GCM AES-GCM AES-GCM AES-KW AES-KW AES-KW AES-KW AES-KW AES-KW AES-KWP AES-KWP AES-KWP AES-KWP AES-KWP
Authenticated Symmetric Decryption	BC-Auth	Authenticated Symmetric Decryption	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-KW AES-KW AES-KW AES-KW AES-KW AES-KW AES-KWP AES-KWP AES-KWP AES-KWP AES-KWP
Key wrapping with AES-CCM	KTS-Wrap	Key wrapping with AES CCM (as permitted by IG D.G)	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM

Name	Type	Description	Properties	Algorithms
				AES-CCM AES-CCM
Key unwrapping with AES-CCM	BC-Auth	Key unwrapping with AES CCM (as permitted by IG D.G)	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM
Key wrapping with AES-GCM	KTS-Wrap	Key wrapping with AES GCM (as permitted by IG D.G)	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM
Key unwrapping with AES-GCM	BC-Auth	Key unwrapping with AES GCM (as permitted by IG D.G)	Key:128, 192, 256 bit keys with 128, 192, 256 bits of key strength, respectively	AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM
Key Pair Generation with ECDSA	AsymKeyPair-KeyGen	Key Pair Generation using ECDSA	ECDSA KeyGen (FIPS186-4):P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA

Name	Type	Description	Properties	Algorithms
				KeyGen (FIPS186-4)
Key Pair Generation with RSA	AsymKeyPair-KeyGen	Key Pair Generation using RSA	RSA KeyGen (FIPS186-4):2048-15360 bits (112-256 bits)	RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4)
Key Pair Generation with Safe Primes	AsymKeyPair-KeyGen	Key Pair Generation using Safe Primes	Safe Primes Key Generation:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 (112-200 bits)	Safe Primes Key Generation
Key Pair Verification with ECDSA	AsymKeyPair-KeyVer	Key Pair Verification using ECDSA	Curves:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits)	ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4)
Key Pair Verification with Safe Primes	AsymKeyPair-KeyVer	Key Pair Verification using Safe Primes	Groups:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 (112-200 bits)	Safe Primes Key Verification
Shared Secret Computation with DH	KAS-SSC	Shared Secret Computation using DH	KAS-FFC-SSC Sp800-56Ar3:ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192,	KAS-FFC-SSC Sp800-56Ar3

Name	Type	Description	Properties	Algorithms
			MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 (112-200 bits strength) Compliance:SP 800-56Arev3, FIPS 140-3 IG D.F. Scenario 2 (1) Scheme:dpEphem KAS Role:initiator, responder	
Shared Secret Computation with ECDH	KAS-SSC	Shared Secret Computation using EC Diffie-Hellman	KAS-ECC-SSC Sp800-56Ar3:P-224, P-256, P-384, P-521 (112, 128, 192, 256 bits strength) Compliance:SP 800-56Arev3, FIPS 140-3 IG D.F. Scenario 2 (1) Scheme:ephemeralUnified KAS Role:initiator, responder	KAS-ECC-SSC Sp800-56Ar3 KAS-ECC-SSC Sp800-56Ar3 KAS-ECC-SSC Sp800-56Ar3 KAS-ECC-SSC Sp800-56Ar3 KAS-ECC-SSC Sp800-56Ar3

Table 8: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES GCM IV

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

The module does not implement the TLS protocol. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

In the event the module's power is lost and restored, the consuming application must ensure that a new key for use with the AES GCM key encryption or decryption under this scenario shall be established.

Alternatively, the Crypto Officer can use the module's API to perform AES GCM encryption using internal IV generation. These IVs are always 96 bits and generated using the approved DRBG internal to the module's boundary, compliant to Scenario 2 of FIPS 140-3 IG C.H.

The module also provides a non-approved AES GCM encryption service which accepts arbitrary external IVs from the operator. This service can be requested by invoking the

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.

Finally, for TLS 1.3, the AES GCM implementation uses the context of Scenario 5 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher-suites that explicitly select AES GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES GCM cipher suites from Section 3.3.1 of SP800-52r2. The module's implementation of AES GCM is used together with an application that runs outside the module's cryptographic boundary. The design of the TLS protocol implicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key.

2.7.2 AES XTS

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

To meet the requirement stated in IG C.I, the module implements a check that ensures, before performing any cryptographic operation, that the two AES keys used in AES XTS mode are not identical.

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

2.7.3 Key Derivation using SP 800-132 PBKDF2

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

- Derived keys shall be used only for storage applications, and shall not be used for any other purposes. The length of the MK or DPK is 112 bits or more.
- Passwords or passphrases, used as an input for the PBKDF2, shall not be used as cryptographic keys.
- The minimum length of the password or passphrase accepted by the module is 8 characters. The probability of guessing the value, assuming a worst-case scenario of all digits, is estimated 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 shall be generated randomly using the SP 800-90Ar1 DRBG provided by the module. The minimum length required is 128 bits.
- The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The minimum value accepted by the module is 1000.

If any of these requirements are not met, the requested service is non-approved (see Non-Approved Services table in Section 4.4 Non-Approved Services).

2.7.4 SP 800-56Ar3 Assurances

To comply with the assurances found in Section 5.6.2 of SP 800-56Ar3, the operator must use the module together with an application that implements the SSH/TLS protocol. Additionally, the module's approved key pair generation service (see Approved Services table in Section 4.3 Approved Services) must be used to generate ephemeral Diffie-Hellman or EC Diffie-Hellman key pairs, or the key pairs must be obtained from another FIPS-validated module. As part of this service, the module will internally perform the full public key validation of the generated public key.

The module's shared secret computation service will internally perform the full public key validation of the peer public key, complying with Sections 5.6.2.2.1 and 5.6.2.2.2 of SP 800-56Ar3.

2.7.5 SHA-3

The module implements the SHA-3 algorithms as both standalone and 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 of SHA-3 include RSA signature generation and verification, ECDSA signature generation and verification, KBKDF, KDA HKDF, X9.63 KDF, X9.42 KDF, PBKDF, OneStep KDA and HMAC. In addition, the implementation of the extendable output functions SHAKE128 and SHAKE256 were verified to have a standalone usage.

2.7.6 RSA Signatures

The module implements only the approved modulus sizes of 2048, 3072, and 4096 bits for signature generation.

For signature verification, the module implements the approved module sizes of 2048, 3072, and 4096 bits. Each algorithm was tested, and corresponding certificates can be found detailed in Section 2.6 Security Function Implementations.

The module also supports RSA signature verification with 1024, 1280, 1536 and 1792 modulus bits. These modulus sizes are allowed only for legacy use, in compliance with FIPS 140-3 IG C.F.

2.8 RBG and Entropy

Cert Number	Vendor Name
E76	Cloudlinux Inc., TuxCare division

Table 9: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Cloudlinux Inc., TuxCare division OpenSSL FIPS	Non-Physical	AlmaLinux 9.2 running on Amazon Web Services (AWS) m5.metal with Intel	64 bits	256 bits	SHA2-512-HMAC-DRBG: A4025; SHA3-256: A4026;

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
provider CPU Time Jitter RNG Entropy Source		Xeon Platinum 8259CL; AlmaLinux 9.2 running on Amazon Web Services (AWS) a1.metal with AWS Graviton			AES-256-CTR-DRBG: A4053

Table 10: Entropy Sources

The module employs two Deterministic Random Bit Generator (DRBG) implementations based on SP 800-90Ar1. These DRBGs are used internally by the module (e.g. to generate seeds for asymmetric key pairs and random numbers for security functions). They can also be accessed using the specified API 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.

These DRBGs will always employ prediction resistance. More information regarding the configuration and design of these DRBGs can be found in the module's manual pages.

2.9 Key Generation

The module implements Cryptographic Key Generation (CKG, vendor affirmed), compliant with SP 800-133r2. When random values are required, they are obtained from the SP 800-90Ar1 approved DRBG, compliant with Section 4 of SP 800-133r2 (without XOR):

- Safe primes key pair generation: compliant with SP 800-133r2, Section 5.2, which maps to SP 800-56Ar3. The method described in Section 5.6.1.1.4 of SP 800-56Ar3 ("Testing Candidates") is used.
- RSA key pair generation: compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-4. The method described in Appendix B.3.6 of FIPS 186-4 ("Probable Primes with Conditions Based on Auxiliary Probable Primes") is used.
- ECC (ECDH and ECDSA) key pair generation: compliant with SP 800-133r2, Section 5.1, which maps to FIPS 186-4. The method described in Appendix B.4.2 of FIPS 186-4 ("Testing Candidates") is used.

Additionally, the module implements the following key derivation methods:

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

- ANS X9.42 KDF, ANS X9.63 KDF: compliant with SP 800-135r1. These implementations shall only be used to generate secret keys in the context of an ANS X9.42-2001 resp. ANS X9.63-2001 key agreement scheme.
- SSH KDF, TLS 1.2 KDF, TLS 1.3 KDF: compliant with SP 800-135r1. These implementations shall only be used to generate 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.

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

2.10 Key Establishment

The module provides Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) shared secret computation compliant with SP800-56Ar3, in accordance with scenario 2 (1) of FIPS 140-3 IG D.F.

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

- 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 curve.

According to FIPS 140-3 IG D.B, the key sizes of DH and ECDH shared secret computation provide 112-200 and 112-256 bits of respective bits of security strength in an approved mode of operation.

The module implements AES KW, KWP, GCM, and CCM as approved key wrapping algorithms. These algorithms can be used to wrap SSPs providing 128, 192, 256-bit keys with key strength between 128-256 bits in compliance with IG D.G. In addition, AES KW and AES KWP meets the requirements of SP 800-38F.

Each approved key wrapping algorithm was tested, and corresponding certificates can be found detailed in Section 2.6 Security Function Implementations.

2.11 Industry Protocols

The module implements the SSH key derivation function 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.

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

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs	Data Input	API input parameters
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs	Data Output	API output parameters
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs	Control Input	API function calls
As a software-only module, the module does not have physical ports. Physical ports are interpreted to be the physical ports of the hardware platform on which it runs	Status Output	API return codes, error queue

Table 11: Ports and Interfaces

The logical interfaces are the APIs through which the applications request services. These logical interfaces are logically separated from each other by the API design. The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not support authentication methods.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 12: Roles

The module supports the Crypto Officer role only. This sole role is implicitly and always assumed by the operator of the module 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
Symmetric Encryption	Used to perform symmetric encryption of an entry plaintext	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Plaintext, AES key	Ciphertext	Symmetric Encryption with AES	Crypto Officer - AES key: W,E
Symmetric Decryption	Used to perform symmetric decryption of an entry ciphertext	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Ciphertext, AES key	Plaintext	Symmetric Decryption with AES	Crypto Officer - AES key: W,E
Authenticated Encryption	Used to perform authenticated symmetric encryption with AES	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Plaintext, AES key, IV	Ciphertext, Tag	Authenticated Symmetric Encryption Key wrapping with AES-KW Key wrapping with AES-KWP Key wrapping with AES-GCM	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Key wrapping with AES-CCM	
Authenticated Decryption	Used to perform authenticated symmetric decryption with AES	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Ciphertext, AES key, IV, Tag	Plaintext	Authenticated Symmetric Decryption Key unwrapping with AES-KW Key unwrapping with AES-KWP Key unwrapping with AES-GCM Key unwrapping with AES-CCM	Crypto Officer - AES key: W,E
Key Wrapping	Perform AES-based key wrapping (compliant to SP800-38F and FIPS 140-3 IG D.G)	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Key to be wrapped, AES key	Wrapped key	Key wrapping with AES-KW Key wrapping with AES-KWP Key wrapping with AES-CCM Key wrapping with AES-GCM	Crypto Officer - AES key: W,E
Key Unwrapping	Perform AES-based key unwrapping (compliant to SP 800-38F and FIPS 140-3 IG D.G)	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_CIPHER_REDHAT_FIPS_INDICATOR_APPROVED	Wrapped key, AES key	Unwrapped key	Key unwrapping with AES-KW Key wrapping with AES-KWP Key wrapping with AES-CCM Key	Crypto Officer - AES key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					wrapping with AES-GCM	
Message Authentication Code	Compute a MAC tag	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_MAC_REDHAT_FIPS_INDICATOR_APPROVED	Message, AES key or HMAC key	MAC tag	Message Authentication Code	Crypto Officer - HMAC key: W,E - AES key: W,E
Message Authentication Code Verification	Verify a MAC tag	OSSL_RH_FIPSINDICATOR_APPROVED, EVP_MAC_REDHAT_FIPS_INDICATOR_APPROVED	Message, AES key or HMAC key, MAC tag	Pass/fail	Message Authentication Code	Crypto Officer - HMAC key: W,E - AES key: W,E
Message Digest	Used to generate a SHA-1, SHA-2, or SHA-3/SHAKE message digest	OSSL_RH_FIPSINDICATOR_APPROVED	Message	Message digest	Message digest	Crypto Officer
Key Derivation with KBKDF	Derive a key from a key-derivation key	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Key-derivation key	KBKDF Derived Key	Key Derivation with KBKDF	Crypto Officer - Key Derivation Key: W,E - KBKDF Derived Key: G,R
Key Derivation with HKDF	Derive a key from a shared secret using HKDF	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	HKDF Derived Key	Key Derivation with HKDF	Crypto Officer - Shared Secret: W,E - HKDF Derived Key: G,R
Key Derivation	Derive a key from a shared	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_	Shared secret	SSH Derived Key	Key Derivation	Crypto Officer - Shared Secret:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
with SSH KDF	secret using SSH KDF	FIPS_INDICATOR_APPROVED			with SSH KDF	W,E - SSH Derived Key: G,R
Key Derivation with X9.63 KDF	Derive a key from a shared secret using X9.63 KDF	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	X9.63 Derived Key	Key Derivation with X9.63 KDF	Crypto Officer - Shared Secret: W,E - X9.63 Derived Key: G,R
Key Derivation with X9.42 KDF	Derive a key from a shared secret using X9.63 KDF	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	X9.42 Derived Key	Key Derivation with X9.42 KDF	Crypto Officer - Shared Secret: W,E - X9.42 Derived Key: G,R
Key Derivation with KDA OneStep	Derive a key from a shared secret using KDA OneStep	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	KDA OneStep Derived Key	Key Derivation with KDA OneStep	Crypto Officer - Shared Secret: W,E - KDA OneStep Derived Key: G,R
TLS Key Derivation	Derive a key from a shared secret using TLS 1.2 KDF / TLS 1.3 KDF	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Shared secret	TLS Derived Key	TLS Key Derivation	Crypto Officer - Shared Secret: W,E - TLS Derived Key: G,R
Password-based Key Derivation	Derive a key from a password	OSSL_RH_FIPSINDICATOR_APPROVED EVP_KDF_REDHAT_FIPS_INDICATOR_APPROVED	Password or passphrase	PBKDF Derived Key	Password-based Key Derivation	Crypto Officer - Password or passphrase:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W,E - PBKDF Derived Key: G,R
Shared Secret Computation	Compute a shared secret	OSSL_RH_FIPSINDICATOR_APPROVED	DH private key, DH public key; EC private key, EC public key	Shared secret	Shared Secret Computation with DH Shared Secret Computation with ECDH	Crypto Officer - DH Private key: W,E - EC Public key: W,E - Shared Secret: G,R - DH Public key : W,E - Shared Secret: G,R
Signature Generation	Generate a digital signature	OSSL_RH_FIPSINDICATOR_APPROVED OSSL_SIGNATURE_PARAM_REDHAT_FIPS_INDICATOR	Message, private key	Signature	Signature Generation	Crypto Officer - RSA private key: W,E - EC Private key: W,E
Signature Verification	Verify a digital signature	OSSL_RH_FIPSINDICATOR_APPROVED OSSL_SIGNATURE_PARAM_REDHAT_FIPS_INDICATOR	Message, public key, signature	Pass/fail	Signature Verification	Crypto Officer - RSA public key: W,E - EC Public key: W,E
Key Pair Generation	Generate a key pair	OSSL_RH_FIPSINDICATOR_APPROVED	Group; Curve; Modulus bits	DH key pair; EC key pair; RSA key pair	Key Pair Generation with RSA Key Pair Generation with ECDSA	Crypto Officer - RSA private key: G,R - DH

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Key Pair Generation with Safe Primes	Private key: G,R - RSA public key: G,R - DH Public key : G,R - EC Private key: G,R - EC Public key: G,R - Intermediate key generation value: G,E,Z
Key Pair Verification	Verify a key pair	OSSL_RH_FIPSINDICATOR_APPROVED	Key pair	Pass/fail	Key Pair Verification with Safe Primes Key Pair Verification with ECDSA	Crypto Officer - DH Public key : W,E - EC Public key: W,E - DH Private key: W,E - EC Private key: W,E
Random Number Generation	Generate random bytes	OSSL_RH_FIPSINDICATOR_APPROVED	Output length	Random bytes	Random Number Generation	Crypto Officer - Entropy input: G,E - DRBG internal state (V value, C value): G,E,W - DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						internal state (V value, Key): G,E,W - DRGB seed: G,E,W
Show status	Show the current status of the module	None	N/A	Module status	None	Crypto Officer
Show module name and version	Show module name and the version of the module	None	N/A	Name and version information	None	Crypto Officer
Self-test	Perform CASTs and integrity test	None	N/A	Pass/fail result of self-tests	Message digest Message Authentication Code Symmetric Encryption with AES Symmetric Decryption with AES Authenticated Symmetric Encryption Authenticated Symmetric Decryption Signature Generation Signature Verification Key Derivation with KBKDF Key Derivation with KDA OneStep Key Derivation with HKDF Key Derivation	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					with X9.42 KDF Key Derivation with X9.63 KDF Key Derivation with SSH KDF TLS Key Derivation Password-based Key Derivation Random Number Generation Shared Secret Computation with DH Shared Secret Computation with ECDH	
Zeroization	Zeroize SSPs.	None	Any SSP	N/A	None	Crypto Officer - AES key: Z - HMAC key: Z - RSA private key: Z - RSA public key: Z - DH Private key: Z - DH Public key : Z - EC Private key: Z - EC Public key: Z - Key Derivation Key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- KBKDF Derived Key: Z - Password or passphrase: Z - PBKDF Derived Key: Z - KBKDF Derived Key: Z - HKDF Derived Key: Z - SSH Derived Key: Z - X9.42 Derived Key: Z - X9.63 Derived Key: Z - KDA OneStep Derived Key: Z - TLS Derived Key: Z - Shared Secret: Z - Entropy input: Z - DRGB seed: Z - DRBG internal state (V value, C value): Z - DRBG internal state (V value, Key): Z - Intermediate

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						key generati on value: Z

Table 13: Approved Services

The module provides services to operators that assume the available role. All services are described in detail in the API documentation (manual pages). The 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.
- **N/A**: The module does not access any SSP or key during its operation.

To interact with the module, a calling application must use the EVP API layer provided by OpenSSL. This layer will delegate the request to the FIPS provider, which will in turn perform the requested service. Additionally, this EVP API layer can be used to retrieve the approved service indicator for the module. The redhat_ossl_query_fipsindicator() function indicates whether an EVP API function is approved.

The exact process to use this function and how to interpret its results is described in the fips_module_indicators manual page.

4.4 Non-Approved Services

Name	Description	Algorithms	Role
AES GCM (external IV)	Authenticated Encryption	AES GCM (external IV)	CO
HMAC (< 112-bit keys)	Compute a MAC tag	HMAC (< 112-bit keys)	CO
Key derivation	Derive a key from a key-derivation key or a shared secret	KBKDF, KDA OneStep, HKDF, ANS X9.42 KDF, ANS X9.63 KDF (< 112-bit keys) KDA OneStep (SHAKE128, SHAKE256) ANS X9.42 KDF (SHAKE128, SHAKE256) ANS X9.63 KDF (SHA-1, SHAKE128, SHAKE256) SSH KDF (SHA-512/224, SHA-512/256, SHA-3, SHAKE128, SHAKE256)	CO

Name	Description	Algorithms	Role
		TLS 1.2 KDF (SHA-1, SHA-224, SHA-512/224, SHA-512/256, SHA-3) TLS 1.3 KDF (SHA-1, SHA-224, SHA-512, SHA-512/224, SHA-512/256, SHA-3)	
PBKDF2 (< 112-bit keys)	Derive a key from a password	PBKDF2 (< 8 characters password; < 128 salt length; < 1000 iterations; < 112-bit keys)	CO
Signature generation	Generate a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO
Signature verification	Verify a signature	RSA and ECDSA (pre-hashed message) RSA-PSS (invalid salt length)	CO
Asymmetric encryption	Encrypt a plaintext	RSA-OAEP	CO
Asymmetric decryption	Decrypt a ciphertext	RSA-OAEP	CO
Shared secret computation	Compute a shared secret	RSA (KAS1, KAS2 schemes)	CO

Table 14: Non-Approved Services

The table above lists the non-approved services in this module, the algorithms involved and the roles that can request the service. In this table, CO specifies the Crypto Officer role.

4.5 External Software/Firmware Loaded

The module does not load external software or firmware.

5 Software/Firmware Security

5.1 Integrity Techniques

The integrity of the module is verified by comparing a HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the fips.so file that was computed at build time. This operation is performed by the verify_integrity() function which performs a KAT for the HMAC SHA-256 algorithm in order to test its proper operation before performing the checksum of the module file.

5.2 Initiate on Demand

Integrity tests are performed as part of the pre-operational self-tests, which are executed when the module is initialized. The integrity test may be invoked on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. This will perform (among others) the software integrity test.

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.

6.2 Configuration Settings and Restrictions

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

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 (N/A).

7.1 Mechanisms and Actions Required

N/A for this module.

7.2 User Placed Tamper Seals

Number: Not applicable.

Placement: Not applicable.

Surface Preparation: Not applicable.

Operator Responsible for Securing Unused Seals: Not applicable.

Part Numbers: Not applicable.

7.3 Filler Panels

Not applicable.

7.4 Fault Induction Mitigation

Not applicable.

7.5 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature			
HighTemperature			
LowVoltage			
HighVoltage			

Table 15: EFP/EFT Information

Not applicable.

7.6 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	
HighTemperature	

Table 16: Hardness Testing Temperatures

Not applicable.

8 Non-Invasive Security

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

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Temporary storage for SSPs used by the module as part of service execution. SSPs are stored until they are zeroized by the operator (using a zeroization call or removing power from the module) or zeroized automatically	Dynamic

Table 17: Storage Areas

The module does not perform persistent storage of SSPs. The SSPs are temporarily stored in the RAM in plaintext form.

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 18: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Free cipher handle	Zeroizes the SSPs contained within the cipher handle.	By calling the appropriate zeroization functions: AES key: EVP_CIPHER_CTX_free and EVP_MAC_CTX_free; HMAC key: EVP_MAC_CTX_free; Key-derivation key: EVP_KDF_CTX_free; Shared secret: EVP_KDF_CTX_free; Password: EVP_KDF_CTX_free; KBKDF Derived Key: EVP_KDF_CTX_free; HKDF Derived Key: EVP_KDF_CTX_free; TLS Derived Key: EVP_KDF_CTX_free; SSH Derived Key: EVP_KDF_CTX_free; X9.63 Derived Key: EVP_KDF_CTX_free; X9.42 Derived Key: EVP_KDF_CTX_free; PBKDF Derived Key: EVP_KDF_CTX_free; KDA OneStep Derived	By calling the cipher related zeroization API

Zeroization Method	Description	Rationale	Operator Initiation
		Key: EVP_KDF_CTX_free; Entropy input: EVP_RAND_CTX_free; DRBG seed: EVP_RAND_CTX_free; DRBG internal state (V value, Key), DRBG internal state (V value, C value): EVP_RAND_CTX_free; DH public & private key: EVP_PKEY_free; EC public & private key: EVP_PKEY_free; RSA public & private key: EVP_PKEY_free	
Automatic	Automatically zeroized by the module when no longer needed	Memory occupied by SSPs is overwritten with zeroes, which renders the SSP values irretrievable.	N/A
Module Reset	De-allocates the volatile memory used to store SSPs	Volatile memory used by the module is overwritten within nanoseconds when power is removed.	By unloading and reloading the module

Table 19: SSP Zeroization Methods

The application that uses the module is responsible for the appropriate zeroization of SSPs. The module provides key allocation and destruction functions, which overwrites the memory occupied by the SSP´s information with zeros before its deallocation.

Memory allocation of SSPs is performed by the OPENSSL_malloc() API call and the application in use of the module is responsible for the calling of the appropriate zeroization functions from the OpenSSL API. The zeroization functions then overwrite the memory occupied by SSPs and de-allocate the memory with the OPENSSL_free() call. OPENSSL_cleanse() should be used to overwrite sensitive data such as private keys.

In case of abnormal termination, or swap in/out of a physical memory page of a process, the SSPs in physical memory are overwritten by the Linux kernel before the physical memory is allocated to another process.

All data output is inhibited during zeroization.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key used for encryption, decryption, and computing MAC tags	AES-XTS: 256, 512 bits; Other modes: 128, 192, 256 bits - AES-XTS: 128, 256 bits; Other modes:	Symmetric key - CSP			Symmetric Encryption with AES Symmetric Decryption with AES Key wrapping with AES-KW Key wrapping

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		128, 192, 256 bits				with AES-KWP Key wrapping with AES-GCM Key wrapping with AES-CCM Key unwrapping with AES-KW Key unwrapping with AES-KWP Key unwrapping with AES-GCM Key unwrapping with AES-CCM Authenticated Symmetric Encryption Authenticated Symmetric Decryption
HMAC key	HMAC key used for computing MAC tags	112-524288 bits - 112-256 bits	Symmetric key - CSP			Message Authentication Code
RSA private key	RSA private key	2048-16384 bits - 112-256 bits	Private key - CSP	Key Pair Generation with RSA		Signature Generation Key Pair Generation with RSA
RSA public key	RSA public key	Signature verification: 1024-16384 bits; Key pair generation: 2048-16384 bits - Signature verification: 80-256	Public key - PSP	Key Pair Generation with RSA		Signature Verification Key Pair Generation with RSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
		bits; Key pair generation: 112-256 bits				
DH Private key	DH Private key	2048-8192 bits - 112-200 bits	Private key - CSP	Key Pair Generation with Safe Primes		Shared Secret Computation with DH Key Pair Generation with Safe Primes Key Pair Verification with Safe Primes
DH Public key	DH Public key	2048-8192 bits - 112-200 bits	Public key - PSP	Key Pair Generation with Safe Primes		Shared Secret Computation with DH Key Pair Generation with Safe Primes Key Pair Verification with Safe Primes
EC Private key	EC Private key used by ECDSA and ECDH	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Private key - CSP	Key Pair Generation with ECDSA		Shared Secret Computation with ECDH Signature Generation Key Pair Verification with ECDSA
EC Public key	EC Public key used by ECDSA and ECDH	P-224, P-256, P-384, P-521 bits - 112, 128, 192, 256 bits	Public key - PSP	Key Pair Generation with ECDSA		Signature Verification Shared Secret Computation with ECDH Key Pair Verification with ECDSA
Key Derivation Key	Symmetric key used to derive symmetric keys	112-4096 bits - 112-256 bits	Symmetric key - CSP			Key Derivation with KBKDF

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KBKDF Derived Key	Symmetric key derived from a key-derivation key	112-4096 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KBKDF		Key Derivation with KBKDF
HKDF Derived Key	Symmetric key derived from a shared secret	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with HKDF		Key Derivation with HKDF
SSH Derived Key	Symmetric key derived from a shared secret	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with SSH KDF		Key Derivation with SSH KDF
X9.63 Derived Key	Symmetric key derived from a shared secret	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with X9.63 KDF		Key Derivation with X9.63 KDF
X9.42 Derived Key	Symmetric key derived from a shared secret	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with X9.42 KDF		Key Derivation with X9.42 KDF
Password or passphrase	Password or passphrase used by PBKDF to derive symmetric keys	8-128 characters - N/A	Password - CSP			Password-based Key Derivation
PBKDF Derived Key	Key derived from PBKDF password/passphrase during key derivation	112-4096 bits - 112-256 bits	Symmetric key - CSP	Password-based Key Derivation		Password-based Key Derivation
KDA OneStep Derived Key	Symmetric key derived from a shared secret	224-8192 bits - 112-256 bits	Symmetric key - CSP	Key Derivation with KDA OneStep		Key Derivation with KDA OneStep
TLS Derived Key	Derived key used in Transport Layer Security (TLS) network protocol	224-8192 bits - 112-256 bits	Symmetric key - CSP	TLS Key Derivation		
Shared Secret	Shared secret generated by ECDH/DH shared	224-8912 bits - 112-256 bits	Shared Secret - CSP		Shared Secret Computation with DH	Shared Secret Computation with DH

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	secret computation				Shared Secret Computation with ECDH	Shared Secret Computation with ECDH Key Derivation with KDA OneStep Key Derivation with HKDF Key Derivation with SSH KDF TLS Key Derivation Key Derivation with X9.63 KDF Key Derivation with X9.42 KDF
Entropy input	Entropy input string used to seed the DRBG (IG D.L compliant)	128-384 bits - 128-384 bits	Entropy Input - CSP			Random Number Generation
DRGB seed	DRBG seed derived from entropy input (IG D.L compliant)	CTR_DRBG: 256, 320, 348 bits; Hash_DRBG: 440, 888 bits; HMAC_DRBG: 160, 256, 512 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG, Hash_DRBG: 128, 256	Seed - CSP	Random Number Generation		Random Number Generation
DRBG internal state (V value, C value)	Internal state of the Hash_DRBG (IG D.L compliant)	880, 1776 bits - 128, 256 bits	Internal state - CSP	Random Number Generation		Random Number Generation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG internal state (V value, Key)	Internal state of the CTR_DRBG and HMAC_DRBG (IG D.L compliant)	CTR_DRBG: 256, 320, 348 bits; HMAC_DRBG: 320, 512, 1024 bits - CTR_DRBG: 128, 192, 256 bits; HMAC_DRBG: 128, 256	Internal state - CSP	Random Number Generation		Random Number Generation
Intermediate key generation value	Intermediate key pair generation value generated during key generation and key derivation services (SP 800-133r2 Section 4, 5.1, and 5.2)	112-256 - 112-256 bits	Intermediate value - CSP	Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes		Key Pair Generation with RSA Key Pair Generation with ECDSA Key Pair Generation with Safe Primes

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	
HMAC key	API input parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	
RSA private key	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	RSA Public key:Paired With Intermediate key generation value:Generated From
RSA public key	API input parameters	RAM:Plaintext	From service invocation until	Free cipher handle	RSA private key:Paired With Intermediate key

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API output parameters		cipherhandle is freed	Module Reset	generation value:Generated From
DH Private key	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	DH Public key:Paired With Intermediate key generation value:Generated From
DH Public key	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	DH Private key:Paired With Intermediate key generation value:Generated From
EC Private key	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	EC Public key:Paired With Intermediate key generation value:Generated From
EC Public key	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	EC private key:Paired With Intermediate key generation value:Generated From
Key Derivation Key	API input parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	KBKDF Derived Key:Derives
KBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Key-derivation key:Derived From
HKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Shared secret:Derived From
SSH Derived Key	API output parameters	RAM:Plaintext	From service invocation until	Free cipher handle Module Reset	Shared secret:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			cipherhandle is freed		
X9.63 Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Shared secret:Derived From
X9.42 Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Shared secret:Derived From
Password or passphrase	API input parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	PBKDF Derived Key:Derives
PBKDF Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Password or passphrase:Derived From
KDA OneStep Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Shared secret:Derived From
TLS Derived Key	API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	Shared secret:Derived From
Shared Secret	API input parameters API output parameters	RAM:Plaintext	From service invocation until cipherhandle is freed	Free cipher handle Module Reset	DH private key:Established By DH public key:Established By EC private key:Established By EC public key:Established By HKDF Derived Key:Derives KDA OneStep Derived Key:Derives TLS Derived Key:Derives SSH Derived Key:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
					X9.63 Derived Key:Derives X9.42 Derived Key:Derives
Entropy input		RAM:Plaintext	From generation until DRBG seed is created	Automatic Module Reset	DRBG seed:Derives
DRGB seed		RAM:Plaintext	While the DRBG is instantiated	Automatic Module Reset	Entropy input:Derived From DRBG internal state (V value, C value):Generates DRBG internal state (V value, Key):Generates
DRBG internal state (V value, C value)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Module Reset	DRBG seed:Generated From
DRBG internal state (V value, Key)		RAM:Plaintext	From DRBG instantiation until DRBG termination	Free cipher handle Module Reset	DRBG seed:Generated From
Intermediate key generation value		RAM:Plaintext	From service invocation until cipherhandle is freed	Automatic	DH private key:Generates DH public key:Generates EC private key:Generates EC public key:Generates RSA private key:Generates RSA public key:Generates

Table 21: 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, 2030.

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A4060)	256-bits key	Message authentication	SW/FW Integrity	Module becomes operational and services are available for use	Integrity test of the shared library component of the module. Verified by comparing an HMAC SHA-256 value calculated at run time with the HMAC SHA-256 value embedded in the fips.so file that was computed at build time.

Table 22: Pre-Operational Self-Tests

The pre-operational software integrity tests are performed automatically when the module is initialized, before the module transitions into the operational state. While the module is executing the self-tests, services are not available, and data output (via the data output interface) is inhibited until the tests are successfully completed. The module transitions to the operational state only after the pre-operational self-tests are passed successfully.

Prior the first use, a CAST is executed for the algorithms used in the Pre-operational Self-Tests.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA-1 (A4059)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4079)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4080)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
SHA-1 (A4081)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA-1 (A4082)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4059)	Message digest	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4079)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4080)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4081)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA2-512 (A4082)	24-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the integrity test
SHA3-256 (A4055)	32-bit message	KAT	CAST	Module becomes operational	Message digest	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-GCM (A4067)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4068)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4069)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4070)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4071)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4072)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4073)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4074)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-GCM (A4075)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4076)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4077)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4078)	256-bit key and 96-bit IV, encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4067)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4068)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4069)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4070)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-GCM (A4071)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4072)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4073)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4074)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4075)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4076)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4077)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-GCM (A4078)	256-bit key and 96-bit IV, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-ECB (A4054)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4056)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4057)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4058)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4061)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4062)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4063)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4064)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-ECB (A4065)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB (A4066)	128-bit key, decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4059)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4079)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4080)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4081)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigGen (FIPS186-4) (A4082)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4059)	PKCS#1 v1.5 with SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	and 2048-bit key					integrity test
RSA SigVer (FIPS186-4) (A4079)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4080)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4081)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
RSA SigVer (FIPS186-4) (A4082)	PKCS#1 v1.5 with SHA-256 and 2048-bit key	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4055)	SHA-256 and P-224, P-256, P-384, and P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4059)	SHA-256 and P-224, P-256, P-384, and P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4079)	SHA-256 and P-224, P-256, P-384, and P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4080)	SHA-256 and P-224, P-256, P-	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	384, and P-521					integrity test
ECDSA SigGen (FIPS186-4) (A4081)	SHA-256 and P-224, P-256, P-384, and P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigGen (FIPS186-4) (A4082)	SHA-256 and P-224, P-256, P-384, and P-521	KAT	CAST	Module becomes operational	Digital signature generation	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4055)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4059)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4079)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4080)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4081)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A4082)	SHA-256	KAT	CAST	Module becomes operational	Digital signature verification	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
KDF SP800-108 (A4084)	HMAC-SHA2-256 in counter mode and 128-bit input key	KAT	CAST	Module becomes operational	Key Derivation with KBKDF	Test runs at power-on before the integrity test
KDA OneStep SP800-56Cr2 (A4051)	SHA2-224 and 448-bit input secret	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDA HKDF Sp800-56Cr1 (A4052)	SHA2-256 and 48-bit secret	KAT	CAST	Module becomes operational	Shared secret key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4055)	AES-128 KW and SHA-1 and 160-bit input secret	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4059)	AES-128 KW and SHA-1 and 160-bit input secret	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4079)	AES-128 KW and SHA-1 and 160-bit input secret	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4080)	AES-128 KW and SHA-1 and 160-bit input secret	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.42 (A4081)	AES-128 KW and SHA-1	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	and 160-bit input secret					integrity test
KDF ANS 9.42 (A4082)	AES-128 KW and SHA-1 and 160-bit input secret	KAT	CAST	Module becomes operational	ANS X9.42 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4055)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4059)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4079)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4080)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4081)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF ANS 9.63 (A4082)	SHA2-256 and 192-bit input secret	KAT	CAST	Module becomes operational	ANS X9.63 key derivation	Test runs at power-on before the integrity test
KDF SSH (A4054)	SHA-1 and 1056-bit input secret	KAT	CAST	Module becomes operational	SSH KDF key derivation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
KDF SSH (A4064)	SHA-1 and 1056-bit input secret	KAT	CAST	Module becomes operational	SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A4065)	SHA-1 and 1056-bit input secret	KAT	CAST	Module becomes operational	SSH KDF key derivation	Test runs at power-on before the integrity test
KDF SSH (A4066)	SHA-1 and 1056-bit input secret	KAT	CAST	Module becomes operational	SSH KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4059)	SHA2-256 and 384-bit input secret	KAT	CAST	Module becomes operational	TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4079)	SHA2-256 and 384-bit input secret	KAT	CAST	Module becomes operational	TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4080)	SHA2-256 and 384-bit input secret	KAT	CAST	Module becomes operational	TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4081)	SHA2-256 and 384-bit input secret	KAT	CAST	Module becomes operational	TLS v1.2 KDF key derivation	Test runs at power-on before the integrity test
TLS v1.2 KDF RFC7627 (A4082)	SHA2-256 and 384-bit input secret	KAT	CAST	Module becomes operational	TLS v1.2 KDF key derivation	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
TLS v1.3 KDF (A4052)	SHA2-256, extract and expand modes	KAT	CAST	Module becomes operational	TLS v1.3 KDF key derivation	Test runs at power-on before the integrity test
PBKDF (A4055)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
PBKDF (A4059)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
PBKDF (A4079)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
PBKDF (A4080)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
PBKDF (A4081)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
PBKDF (A4082)	SHA2-256	KAT	CAST	Module becomes operational	Password-based Key Derivation	Test runs at power-on before the integrity test
Counter DRBG (A4053)	AES-128 with derivation function and	KAT	CAST	Module becomes operational	Instantiate; Generate; Reseed (compliant to SP 800-	Test runs at power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	prediction resistance				90Arev1 Section 11.3)	integrity test
Hash DRBG (A4053)	SHA2-256	KAT	CAST	Module becomes operational	Instantiate; Generate; Reseed (compliant to SP 800-90Arev1 Section 11.3)	Test runs at power-on before the integrity test
HMAC DRBG (A4053)	HMAC-SHA-1	KAT	CAST	Module becomes operational	Instantiate; Generate; Reseed (compliant to SP 800-90Arev1 Section 11.3)	Test runs at power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A4085)	ffdhe2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4059)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4079)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4080)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A4081)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3 (A4082)	P-256	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at power-on before the integrity test
Safe Primes Key Generation (A4085)	N/A	PCT	PCT	Key pair generation is successful	SP 800-56Arev3 Section 5.6.2.1.4	Key pair generation
RSA KeyGen (FIPS186-4) (A4059)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4079)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4080)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4081)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
RSA KeyGen (FIPS186-4) (A4082)	PKCS#1 v1.5 with SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4059)	SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4079)	SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
ECDSA KeyGen	SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
(FIPS186-4) (A4080)						
ECDSA KeyGen (FIPS186-4) (A4081)	SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation
ECDSA KeyGen (FIPS186-4) (A4082)	SHA-256	PCT	PCT	Key pair generation is sucessful	Signature generation and verification	Key pair generation

Table 23: Conditional Self-Tests

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

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A4060)	Message authentication	SW/FW Integrity	On demand	Manually

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA-1 (A4059)	KAT	CAST	On Demand	Manually
SHA-1 (A4079)	KAT	CAST	On Demand	Manually
SHA-1 (A4080)	KAT	CAST	On Demand	Manually
SHA-1 (A4081)	KAT	CAST	On Demand	Manually
SHA-1 (A4082)	KAT	CAST	On Demand	Manually
SHA2-512 (A4059)	KAT	CAST	On Demand	Manually
SHA2-512 (A4079)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA2-512 (A4080)	KAT	CAST	On Demand	Manually
SHA2-512 (A4081)	KAT	CAST	On Demand	Manually
SHA2-512 (A4082)	KAT	CAST	On Demand	Manually
SHA3-256 (A4055)	KAT	CAST	On Demand	Manually
AES-GCM (A4067)	KAT	CAST	On Demand	Manually
AES-GCM (A4068)	KAT	CAST	On Demand	Manually
AES-GCM (A4069)	KAT	CAST	On Demand	Manually
AES-GCM (A4070)	KAT	CAST	On Demand	Manually
AES-GCM (A4071)	KAT	CAST	On Demand	Manually
AES-GCM (A4072)	KAT	CAST	On Demand	Manually
AES-GCM (A4073)	KAT	CAST	On Demand	Manually
AES-GCM (A4074)	KAT	CAST	On Demand	Manually
AES-GCM (A4075)	KAT	CAST	On Demand	Manually
AES-GCM (A4076)	KAT	CAST	On Demand	Manually
AES-GCM (A4077)	KAT	CAST	On Demand	Manually
AES-GCM (A4078)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM (A4067)	KAT	CAST	On Demand	Manually
AES-GCM (A4068)	KAT	CAST	On Demand	Manually
AES-GCM (A4069)	KAT	CAST	On Demand	Manually
AES-GCM (A4070)	KAT	CAST	On Demand	Manually
AES-GCM (A4071)	KAT	CAST	On Demand	Manually
AES-GCM (A4072)	KAT	CAST	On Demand	Manually
AES-GCM (A4073)	KAT	CAST	On Demand	Manually
AES-GCM (A4074)	KAT	CAST	On Demand	Manually
AES-GCM (A4075)	KAT	CAST	On Demand	Manually
AES-GCM (A4076)	KAT	CAST	On Demand	Manually
AES-GCM (A4077)	KAT	CAST	On Demand	Manually
AES-GCM (A4078)	KAT	CAST	On Demand	Manually
AES-ECB (A4054)	KAT	CAST	On Demand	Manually
AES-ECB (A4056)	KAT	CAST	On Demand	Manually
AES-ECB (A4057)	KAT	CAST	On Demand	Manually
AES-ECB (A4058)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB (A4061)	KAT	CAST	On Demand	Manually
AES-ECB (A4062)	KAT	CAST	On Demand	Manually
AES-ECB (A4063)	KAT	CAST	On Demand	Manually
AES-ECB (A4064)	KAT	CAST	On Demand	Manually
AES-ECB (A4065)	KAT	CAST	On Demand	Manually
AES-ECB (A4066)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4059)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4079)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4080)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4081)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A4082)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4059)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4079)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A4080)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4081)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (A4082)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4055)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4059)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4079)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4080)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4081)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A4082)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4055)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4059)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4079)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ECDSA SigVer (FIPS186-4) (A4080)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4081)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A4082)	KAT	CAST	On Demand	Manually
KDF SP800-108 (A4084)	KAT	CAST	On Demand	Manually
KDA OneStep SP800-56Cr2 (A4051)	KAT	CAST	On Demand	Manually
KDA HKDF Sp800-56Cr1 (A4052)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4055)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4059)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4079)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4080)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4081)	KAT	CAST	On Demand	Manually
KDF ANS 9.42 (A4082)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4055)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4059)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
KDF ANS 9.63 (A4079)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4080)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4081)	KAT	CAST	On Demand	Manually
KDF ANS 9.63 (A4082)	KAT	CAST	On Demand	Manually
KDF SSH (A4054)	KAT	CAST	On Demand	Manually
KDF SSH (A4064)	KAT	CAST	On Demand	Manually
KDF SSH (A4065)	KAT	CAST	On Demand	Manually
KDF SSH (A4066)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4059)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4079)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4080)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4081)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A4082)	KAT	CAST	On Demand	Manually
TLS v1.3 KDF (A4052)	KAT	CAST	On Demand	Manually
PBKDF (A4055)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
PBKDF (A4059)	KAT	CAST	On Demand	Manually
PBKDF (A4079)	KAT	CAST	On Demand	Manually
PBKDF (A4080)	KAT	CAST	On Demand	Manually
PBKDF (A4081)	KAT	CAST	On Demand	Manually
PBKDF (A4082)	KAT	CAST	On Demand	Manually
Counter DRBG (A4053)	KAT	CAST	On Demand	Manually
Hash DRBG (A4053)	KAT	CAST	On Demand	Manually
HMAC DRBG (A4053)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A4085)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4059)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4079)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4080)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4081)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A4082)	KAT	CAST	On Demand	Manually
Safe Primes Key Generation (A4085)	PCT	PCT	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Safe Primes Key Generation (A4085)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4059)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4079)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4080)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4081)	PCT	PCT	On Demand	Manually
RSA KeyGen (FIPS186-4) (A4082)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4059)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4079)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4080)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4081)	PCT	PCT	On Demand	Manually
ECDSA KeyGen (FIPS186-4) (A4082)	PCT	PCT	On Demand	Manually

Table 25: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error	If the module fails any of the self-tests, the module enters the error state. In the error state, the module immediately stops functioning and ends the application process	Software integrity test failure CAST failure	Module reinitialization	OSSL_PROV_PARAM_STATUS is set to 0. Module will not load.
PCT Error	Pairwise consistency test failure	PCT failure	Module reinitialization	Module is aborted

Table 26: Error States

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

10.5 Operator Initiation of Self-Tests

Both conditional and pre-operational self-tests can be executed on-demand by unloading and subsequently re-initializing the module, or by calling the OSSL_PROVIDER_self_test function. The pair-wise consistency tests can be invoked on demand by requesting the key pair generation service.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The module is distributed as a part of the AlmaLinux 9 OpenSSL package in the form of the openssl-libs-3.0.7-20.el9_2.tuxcare.1 RPM package.

Before the openssl-libs-3.0.7-20.el9_2.tuxcare.1 RPM package is installed, the AlmaLinux 9 system must operate in the FIPS validated configuration. This can be achieved by switching the system into the FIPS-validated configuration after the installation. Execute the openssl list -providers command. Restart the system.

The Crypto Officer must verify the AlmaLinux 9 system operates in the FIPS-validated configuration by executing the fips-mode-setup –check command, which should output "FIPS mode is enabled."

11.2 Administrator Guidance

After the openssl-libs-3.0.7-20.el9_2.tuxcare.1 RPM package is installed, the Crypto Officer must execute the openssl list -providers command. This command should display the base/default and FIPS providers as follows:

```
Providers
base
        name: OpenSSL Base Provider
        version: 3.0.7
        status: active
default
        name: OpenSSL Default Provider
        version: 3.0.7
        status: active
fips
        name: OpenSSL FIPS Provider for AlmaLinux 9
        version: 3.0.7-1d2bd88ee26b3c90
        status: active
```

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

11.3 Non-Administrator Guidance

There is no administrator guidance.

11.4 Design and Rules

Not applicable.

11.5 Maintenance Requirements

Not applicable

11.6 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of unloading the module. This will zeroize all SSPs in volatile memory. Then, if desired, the openssl-libs-3.0.7-20.el9_2.tuxcare.1 RPM package can be uninstalled from the AlmaLinux 9 system

12 Mitigation of Other Attacks

12.1 Attack List

Certain cryptographic subroutines and algorithms are vulnerable to timing analysis. The module claims mitigation of timing-based side-channel attacks implementing two methods: Constant-time Implementations and Numeric Blinding:

- Constant-time Implementations protect cryptographic implementations in the module against timing cryptanalysis ensuring that the variations in execution time for different cryptographic algorithms cannot be traced back to the key, CSP or secret data.
- Numeric Blinding protects the RSA and ECDSA algorithms from timing attacks. These algorithms are vulnerable to such attacks since attackers can measure the time of signature operations or RSA decryption. To mitigate this, the module generates a random factor which is provided as an input to the decryption/signature operation which discarded once the operation results in an output. This makes it difficult for attackers to attempt timing attacks making impossible correlating execution time to the RSA/ECDSA key.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
CAST	Cryptographic Algorithm Self-Test
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CKG	Cryptographic Key Generation
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EVP	Envelope
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GMAC	Galois Counter Mode Message Authentication Code
HKDF	HMAC-based Key Derivation Function
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
KAS	Key Agreement Scheme
KAT	Known Answer Test
KBKDF	Key-based Key Derivation Function
KW	Key Wrap
KWP	Key Wrap with Padding
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PCT	Pair-wise Consistency Test
PBKDF2	Password-based Key Derivation Function v2
PSS	Probabilistic Signature Scheme
RSA	Rivest, Shamir, Adleman

SHA	Secure Hash Algorithm
SSC	Shared Secret Computation
SSH	Secure Shell
SSP	Sensitive Security Parameter
TLS	Transport Layer Security
XOF	Extendable Output Function
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

ANS X9.42-2001 **Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography**

2001

https://webstore.ansi.org/standards/ascx9/ansix9422001

ANS X9.63-2001 **Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography**

2001

https://webstore.ansi.org/standards/ascx9/ansix9632001

FIPS 140-3 **FIPS PUB 140-3 - Security Requirements for Cryptographic Modules**

March 2019

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf

FIPS 140-3 IG **Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program**

March 2024

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://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

FIPS 186-4 **Digital Signature Standard (DSS)**

July 2013

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

FIPS 197 **Advanced Encryption Standard**

November 2001

https://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS 198-1 **The Keyed Hash Message Authentication Code (HMAC)**

July 2008

https://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

FIPS 202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
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 800-140B	**NIST Special Publication 800-140B - CMVP Security Policy Requirements** March 2020 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140B.pdf
SP 800-38A	**Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf
SP 800-38A Addendum	**Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode** October 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf

SP 800-38B **Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication**

May 2005

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf

SP 800-38C **Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality**

July 2007

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf

SP 800-38D **Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC**

November 2007

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

SP 800-38E **Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices**

January 2010

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf

SP 800-38F **Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping**

December 2012

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf

SP 800-52r2 **Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations**

August 2019

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

SP 800-56Ar3 **Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography**

April 2018

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf

SP 800-56Cr1 **Recommendation for Key-Derivation Methods in Key-Establishment Schemes**

April 2018

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr1.pdf

SP 800-56Cr2 **Recommendation for Key-Derivation Methods in Key-Establishment Schemes**

August 2020

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Cr2.pdf

SP 800-90Ar1 **Recommendation for Random Number Generation Using Deterministic Random Bit Generators**

June 2015

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

SP 800-90B **Recommendation for the Entropy Sources Used for Random Bit Generation**

January 2018

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf

SP 800-108r1 **NIST Special Publication 800-108r1 - Recommendation for Key Derivation Using Pseudorandom Functions**

August 2022

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1.pdf

SP 800-132 **Recommendation for Password-Based Key Derivation - Part 1: Storage Applications**

December 2010

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf

SP 800-133r2 **Recommendation for Cryptographic Key Generation**

June 2020

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf

SP 800-135r1	Recommendation for Existing Application-Specific Key Derivation Functions December 2011 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf