KeyPair Consulting Inc.

KeyPair FIPS Provider for OpenSSL 3

FIPS 140-3 Non-Proprietary Security Policy

Document Version 1.4

March 10, 2025

KeyPair Consulting Inc.
987 Osos Street
San Luis Obispo, CA 93401
keypair.us
+1 805.316.5024

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document defines the Non-Proprietary Security Policy for the *KeyPair FIPS Provider for OpenSSL 3* cryptographic module by KeyPair Consulting Inc., hereafter denoted the Module. The Module meets FIPS 140-3 overall Level 1 requirements, with security levels as shown in Section 1.2. In accordance with AS02.05, ISO/IEC 19790:2012 §7.7 Physical Security is optional and does not apply to the Module.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Module is a cryptographic software library, intended for use by US and Canadian Federal agencies and other markets that require FIPS 140-3 validated cryptographic functionality. The Module is distributed by KeyPair to vendors for incorporation into their products.

The Module design corresponds to the Module security rules. Security rules enforced by the Module are described in the appropriate context of this document.

Module Type: Software

Module Embodiment: MultiChipStand

Cryptographic Boundary:

Figure 1 depicts the Module operational environment, with the cryptographic boundary highlighted in red inclusive of all Module entry points (API calls). The Module is defined as a *Software module* per AS02.03.

The pre-operational approved integrity test is performed over all components within the cryptographic boundary.

Tested Operational Environment's Physical Perimeter (TOEPP):

The General Purpose Computer is the TOEPP.

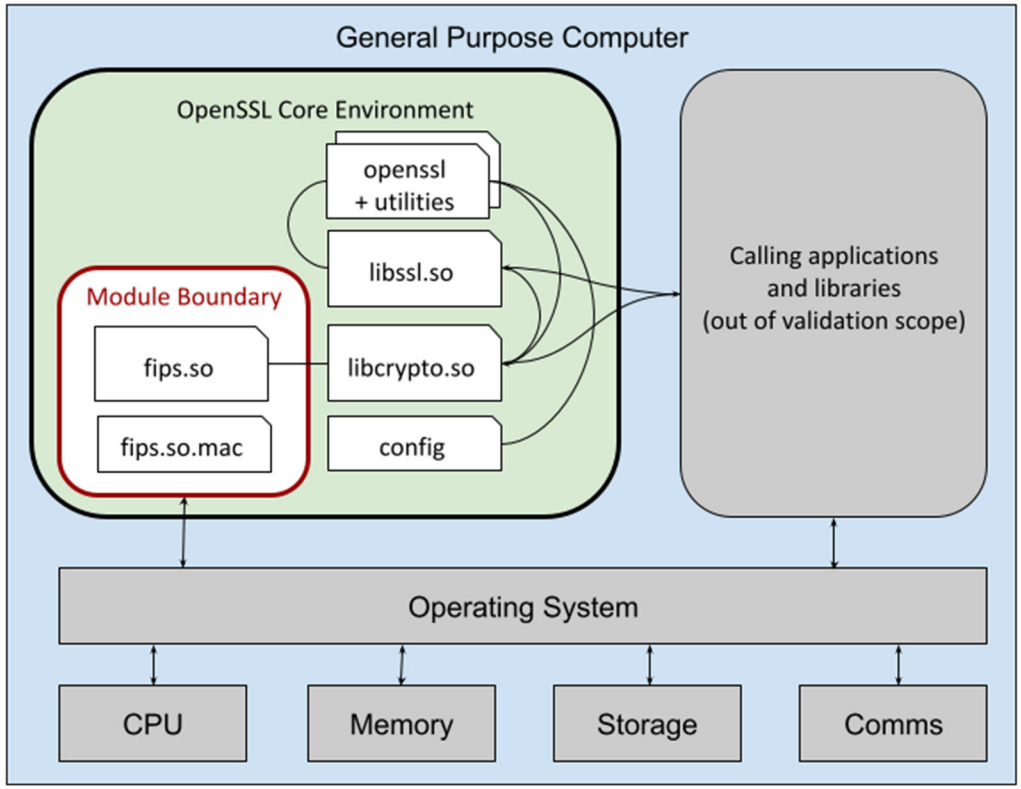

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fips.so	3.0.10 with KP_1.2	N/A	HMAC-SHA2-256 #A4481 over the complete module file image

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
AIX 7.3 32-bit	IBM Power E1080 (9080-HEX)	IBM POWER10 (PPC)	Yes		3.0.10 with KP_1.2
AIX 7.3 32-bit	IBM Power E1080 (9080-HEX)	IBM POWER10 (PPC)	No		3.0.10 with KP_1.2
AIX 7.3 64-bit	IBM Power E1080 (9080-HEX)	IBM POWER10 (PPC)	Yes		3.0.10 with KP_1.2
AIX 7.3 64-bit	IBM Power E1080 (9080-HEX)	IBM POWER10 (PPC)	No		3.0.10 with KP_1.2
HID Intelligent Controller OS 2.0	LP4502	Atmel SAMA5 (ARMv7)	No		3.0.10 with KP_1.2
HID Intelligent Controller OS 2.0	MP4502	STMicroelectronics STM32MP133CAF3 (ARMv7)	No		3.0.10 with KP_1.2
Linux 5.4	Gallagher Controller 7000	NXP i.MX 8X (Cortex-A35)	Yes		3.0.10 with KP_1.2
Linux 5.4	Gallagher Controller 7000	NXP i.MX 8X (Cortex-A35)	No		3.0.10 with KP_1.2
Linux 6.1	Gallagher Controller 6000	Atmel AT91SAM9	No		3.0.10 with KP_1.2
PexOS 1.0	SuperMicro SuperServer 6018R-MTR	Intel® Xeon® Gold 6248 (Cascade Lake)	Yes	VMware ESXi 7	3.0.10 with KP_1.2
PexOS 1.0	SuperMicro SuperServer 6018R-MTR	Intel® Xeon® Gold 6248 (Cascade Lake)	No	VMware ESXi 7	3.0.10 with KP_1.2
Panzura Base OS 1.2	Lenovo ThinkPad X1 Extreme Gen 3	Intel® Core™ i7-10850H vPro® CPU @ 2.70GHz (Comet Lake)	Yes		3.0.10 with KP_1.2
Panzura Base OS 1.2	Lenovo ThinkPad X1 Extreme Gen 3	Intel® Core™ i7-10850H vPro® CPU @ 2.70GHz (Comet Lake)	No		3.0.10 with KP_1.2
Scientific Linux 6.9	SteelHead CXA-00580	Intel® Xeon® D-1513N (Broadwell)	Yes		3.0.10 with KP_1.2
Scientific Linux 6.9	SteelHead CXA-00580	Intel® Xeon® D-1513N (Broadwell)	No		3.0.10 with KP_1.2
SFOS 20.0	XGS 3100	AMD Ryzen Embedded V1780B	Yes		3.0.10 with KP_1.2
SFOS 20.0	XGS 3100	AMD Ryzen Embedded V1780B	No		3.0.10 with KP_1.2
SnowflakeOS 1.2	HPE ProLiant DL60 Gen9	Intel® Xeon® E5-2609 (Sandy Bridge EP)	Yes		3.0.10 with KP_1.2
SnowflakeOS 1.2	HPE ProLiant DL60 Gen9	Intel® Xeon® E5-2609 (Sandy Bridge EP)	No		3.0.10 with KP_1.2
SnowflakeOS 1.2	Ampere® Altra® 2U Server R272-P33	Ampere® Altra® SOC with Aarch64 ARMv8	Yes		3.0.10 with KP_1.2

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
SnowflakeOS 1.2	Ampere® Altra® 2U Server R272-P33	Ampere® Altra® SOC with Aarch64 ARMv8	No		3.0.10 with KP_1.2
Ubuntu 20.04 LTS	Dell Inspiron 7591	Intel Core i7-10510U	Yes		3.0.10 with KP_1.2
Ubuntu 20.04 LTS	Dell Inspiron 7591	Intel Core i7-10510U	No		3.0.10 with KP_1.2
Ubuntu 20.04 LTS	Dell PowerEdge R7515	AMD EPYC 7313P	Yes		3.0.10 with KP_1.2
Ubuntu 20.04 LTS	Dell PowerEdge R7515	AMD EPYC 7313P	No		3.0.10 with KP_1.2
ZPE Systems' Nodegrid OS version 6.0	Net Services Router (NSR)	Intel® Atom™ CPU C3758 (Denverton)	Yes		3.0.10 with KP_1.2
ZPE Systems' Nodegrid OS version 6.0	Net Services Router (NSR)	Intel® Atom™ CPU C3758 (Denverton)	No		3.0.10 with KP_1.2
Red Hat Enterprise Linux 9	HPE ProLiant XL420 Gen10	Intel® Xeon® Silver 4214R CPU	Yes		3.0.10 with KP_1.2
Red Hat Enterprise Linux 9	HPE ProLiant XL420 Gen10	Intel® Xeon® Silver 4214R CPU	No		3.0.10 with KP_1.2
Microsoft Windows 11 Pro	Dell Optiplex SFF 7020	Intel® Core™ i3-14100 (Raptor Lake)	Yes		3.0.10 with KP_1.2
Microsoft Windows 11 Pro	Dell Optiplex SFF 7020	Intel® Core™ i3-14100 (Raptor Lake)	No		3.0.10 with KP_1.2
Microsoft Windows Server 2022	Dell PowerEdge R450	Intel® Xeon® Silver 4309Y (Ice Lake)	Yes		3.0.10 with KP_1.2
Microsoft Windows Server 2022	Dell PowerEdge R450	Intel® Xeon® Silver 4309Y (Ice Lake)	No		3.0.10 with KP_1.2

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

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

Operating System	Hardware Platform
Ubuntu 18.04	Dell Inspiron 7591 with Intel® Core™ i7-10510U
Ubuntu 18.04	Dell PowerEdge R7515 with AMD EPYC 7313P
Ubuntu 22.04 LTS	HPE ProLiant DL325 Gen10 Plus v2 with AMD EPYC 7313P
Ubuntu 22.04 LTS	HPE ProLiant DL60 Gen9 with Intel® Xeon® E5-2609
CentOS 7.9	Ampere® Altra® 2U Server R272-P33 with Ampere® Altra® SOC with Aarch64 ARMv8
CentOS 7.9	HW Platform: HPE ProLiant DL60 Gen9 with Intel® Xeon® E5-2609

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

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

2.3 Excluded Components

N/A for this Module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Nominal	Approved mode of operation	Approved	

Table 5: Modes List and Description

The Module only supports an Approved mode of operation. The conditions for using the Module in the Approved mode of operation are:

1. Installation of the Module as described in Section 11.1 results in the settings described below, which are required for operation in the Approved mode:
    a. security-checks = 1
    Enforce minimum key strengths and approved curve names.
    b. allow-plaintext-csp-output = 1
    Enforce the AS09.16 and AS09.17 requirement for a second independent action to output CSPs as a result of calls that produce CSPs, such as key generation, key unwrap (or decapsulate) and shared secret calculation.
    c. conditional-errors = 1
    Enforce the Module entering the error state on conditional test errors such as PCT failure.
2. The Module is a cryptographic library used by a calling application. The calling application is responsible for:
    a. Use of the primitives in the correct sequence.
    b. Use of keys in accordance with SP 800-140D Rev. 2 (as the keys used by the Module for cryptographic purposes are provided over the call stack by the calling application).
    c. Use of a SP 800-90B compliant entropy source outside the Module boundary with at least 256 bits of security strength. Entropy is supplied to the Module via callback functions. The callback functions shall return an error if the minimum entropy strength cannot be met.

2.5 Algorithms

Approved Algorithms:

Cipher

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS1	A4481	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS2	A4481	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC-CS3	A4481	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A4481	Key Length - 128, 192, 256	SP 800-38C
AES-CFB1	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB8	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CTR	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A4481	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
AES-KW	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KWP	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-OFB	A4481	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-XTS Testing Revision 2.0	A4481	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38E

Table 6: Approved Algorithms - Cipher

Key agreement

Algorithm	CAVP Cert	Properties	Reference
KAS-ECC CDH-Component SP800-56Ar3 (CVL)	A4481	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A4481	Domain Parameter Generation Methods - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A4481	Domain Parameter Generation Methods - FB, FC, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, modp-2048, modp-3072, modp-4096, modp-6144, modp-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3

Algorithm	CAVP Cert	Properties	Reference
KAS-IFC-SSC	A4481	Modulo - 2048, 3072, 4096, 6144, 8192 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Scheme - KAS1 - KAS Role - initiator, responder Scheme - KAS2 - KAS Role - initiator, responder	SP 800-56A Rev. 3

Table 7: Approved Algorithms - Key agreement

Key derivation

Algorithm	CAVP Cert	Properties	Reference
KDA HKDF SP800-56Cr2	A4481	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8 HMAC Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512	SP 800-56C Rev. 2
KDA OneStep SP800-56Cr2	A4481	Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8	SP 800-56C Rev. 2
KDA TwoStep SP800-56Cr2	A4481	MAC Salting Methods - default, random KDF Mode - feedback Derived Key Length - 2048 Shared Secret Length - Shared Secret Length: 224-8192 Increment 8	SP 800-56C Rev. 2
KDF ANS 9.42 (CVL)	A4481	KDF Type - DER Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256, SHA3-224, SHA3-256, SHA3-384, SHA3-512 Key Data Length - Key Data Length: 8-4096 Increment 8	SP 800-135 Rev. 1
KDF ANS 9.63 (CVL)	A4481	Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512 Key Data Length - Key Data Length: 128, 4096	SP 800-135 Rev. 1
KDF SP800-108	A4481	KDF Mode - Counter, Feedback Supported Lengths - Supported Lengths: 8, 72, 128, 776, 3456, 4096	SP 800-108 Rev. 1
KDF SSH (CVL)	A4481	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
PBKDF	A4481	Iteration Count - Iteration Count: 1-10000 Increment 1 Password Length - Password Length: 8-128 Increment 8	SP 800-132
TLS v1.2 KDF RFC7627 (CVL)	A4481	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A4481	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 8: Approved Algorithms - Key derivation

Key management

Algorithm	CAVP Cert	Properties	Reference
DSA KeyGen (FIPS186-4)	A4481	L - 2048, 3072 N - 224, 256	FIPS 186-4
DSA PQGGen (FIPS186-4)	A4481	L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA PQGVer (FIPS186-4)	A4481	L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A4481	Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A4481	Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521	FIPS 186-4
EDDSA KeyGen	A4481	Curve - ED-25519, ED-448	FIPS 186-5
EDDSA KeyVer	A4481	Curve - ED-25519, ED-448	FIPS 186-5
Safe Primes Key Generation	A4481	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, modp-2048, modp-3072, modp-4096, modp-6144, modp-8192	SP 800-56A Rev. 3
Safe Primes Key Verification	A4481	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, modp-2048, modp-3072, modp-4096, modp-6144, modp-8192	SP 800-56A Rev. 3
RSA KeyGen (FIPS186-4)	A4481	Key Generation Mode - B.3.3 Modulo - 2048, 3072, 4096 Primality Tests - Table C.2 Private Key Format - Standard	FIPS 186-4

Table 9: Approved Algorithms - Key management

Key transport

Algorithm	CAVP Cert	Properties	Reference
KTS-IFC	A4481	Modulo - 2048, 3072, 4096, 6144 Key Generation Methods - rsakpg1-basic, rsakpg1-crt, rsakpg1-prime-factor, rsakpg2-basic, rsakpg2-crt, rsakpg2-prime-factor Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Key Length - 1024	SP 800-56B Rev. 2

Table 10: Approved Algorithms - Key transport

Message authentication

Algorithm	CAVP Cert	Properties	Reference
AES-CMAC	A4481	Direction - Generation, Verification Key Length - 128, 192, 256	SP 800-38B
AES-GMAC	A4481	Direction - Decrypt, Encrypt IV Generation - External, Internal IV Generation Mode - 8.2.1 Key Length - 128, 192, 256	SP 800-38D
HMAC-SHA-1	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-224	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-256	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-384	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-512	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-512/224	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA3-224	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA3-256	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA3-384	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
HMAC-SHA3-512	A4481	Key Length - Key Length: 112-2048 Increment 8	FIPS 198-1
KMAC-128	A4481	Message Length - Message Length: 0-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8	SP 800-185
KMAC-256	A4481	Message Length - Message Length: 0-65536 Increment 8 Key Data Length - Key Data Length: 128-1024 Increment 8	SP 800-185

Table 11: Approved Algorithms - Message authentication

Message digest

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-224	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-256	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-384	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-512/224	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA2-512/256	A4481	Message Length - Message Length: 0-65528 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 180-4
SHA3-224	A4481	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-256	A4481	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-384	A4481	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHA3-512	A4481	Message Length - Message Length: 0-65536 Increment 8 Large Message Sizes - 1, 2, 4, 8	FIPS 202
SHAKE-128	A4481	Output Length - Output Length: 16-65536 Increment 8	FIPS 202
SHAKE-256	A4481	Output Length - Output Length: 16-65536 Increment 8	FIPS 202

Table 12: Approved Algorithms - Message digest

Random

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A4481	Prediction Resistance - Yes Mode - AES-128, AES-192, AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
Hash DRBG	A4481	Prediction Resistance - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	SP 800-90A Rev. 1
HMAC DRBG	A4481	Prediction Resistance - Yes Mode - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	SP 800-90A Rev. 1

Table 13: Approved Algorithms - Random

Signature

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigGen (FIPS186-4)	A4481	Component - No, Yes Curve - B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A4481	Component - No Curve - B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
DSA SigGen (FIPS186-4)	A4481	L - 2048, 3072 N - 224, 256 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
DSA SigVer (FIPS186-4)	A4481	L - 1024, 2048, 3072 N - 160, 224, 256 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256	FIPS 186-4
EDDSA SigGen	A4481	Curve - ED-25519, ED-448	FIPS 186-5
EDDSA SigVer	A4481	Curve - ED-25519, ED-448	FIPS 186-5
RSA SigGen (FIPS186-4)	A4481	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-5)	A4481	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA Signature Primitive (CVL)	A4481	Private Key Format - crt	FIPS 186-4
RSA SigVer (FIPS186-4)	A4481	Signature Type - ANSI X9.31, PKCS 1.5, PKCSPSS Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A4481	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5

Table 14: Approved Algorithms - Signature

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG Section 4		KeyPair FIPS Provider for OpenSSL 3	NIST, SP 800-133 Rev. 2
CKG Section 5		KeyPair FIPS Provider for OpenSSL 3	NIST, SP 800-133 Rev. 2
CKG Section 6.2		KeyPair FIPS Provider for OpenSSL 3	NIST, SP 800-133 Rev. 2
Hash DRBG with SHA3-256, SHA3-512		KeyPair FIPS Provider for OpenSSL 3	NIST, SP 800-90A Rev. 1
HMAC DRBG with SHA3-256, SHA3-512		KeyPair FIPS Provider for OpenSSL 3	NIST, SP 800-90A Rev. 1

Table 15: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this Module.

Non-Approved, Not Allowed Algorithms:

N/A for this Module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Cipher (Unauth)	BC-UnAuth	AES ciphers		AES-CBC AES-CBC-CS1 AES-CBC-CS2 AES-CBC-CS3 AES-CFB1 AES-CFB128 AES-CFB8 AES-CTR AES-ECB AES-OFB AES-XTS Testing Revision 2.0
Cipher (Auth)	BC-Auth	Authenticated ciphers		AES-CCM AES-GCM AES-KW AES-KWP
CKG Section 4	CKG	Using the Output of a Random Bit Generator		CKG Section 4
CKG Section 5	CKG	Generation of Key Pairs for Asymmetric-Key Algorithms		CKG Section 5
CKG Section 6.2	CKG	Derivation of Symmetric Keys		CKG Section 6.2
Key agreement	KAS-SSC	Key agreement	KAS:KAS-ECC-SSC provides between 112 and 256 bits of encryption strength; KAS-FFC-SSC provides between 112 and 200 bits of encryption strength; KAS-IFC-SSC provides between 112 and 200 bits of encryption strength	KAS-ECC CDH-Component SP800-56Ar3 KAS-ECC-SSC Sp800-56Ar3 KAS-FFC-SSC Sp800-56Ar3 KAS-IFC-SSC
Key derivation	KAS-135KDF KAS-56CKDF KBKDF PBKDF			KAS-KDF HKDF SP800-56Cr2 KAS-KDF OneStep SP800-56Cr2 KAS-KDF TwoStep SP800-56Cr2 KDF ANS 9.42 KDF ANS 9.63 KDF SP800-108 KDF SSH PBKDF TLS v1.2 KDF RFC7627 TLS v1.3 KDF

Name	Type	Description	Properties	Algorithms
Key management ECC	AsymKeyPair-KeyGen AsymKeyPair-KeyVer			ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)
Key management Edwards	AsymKeyPair-KeyGen AsymKeyPair-KeyVer			EDDSA KeyGen EDDSA KeyVer
Key management FFC	AsymKeyPair-KeyGen			DSA KeyGen (FIPS186-4) DSA PQGGen (FIPS186-4) DSA PQGVer (FIPS186-4) Safe Primes Key Generation Safe Primes Key Verification
Key management IFC	AsymKeyPair-KeyGen			RSA KeyGen (FIPS186-4)
Key transport	KTS-Encap		KTS:2048, 3072, 4096 or 6144-bit keys provide between 112 and 176 bits of encryption strength	KTS-IFC
KTS (Cipher w/ CMAC, GMAC, HMAC, KMAC)	BC-Auth BC-UnAuth MAC	SP 800-38F Section 3.1 Provisions	KTS:128, 192 or 256-bit keys provide between 128 and 256 bits of encryption strength	AES-CBC AES-CBC-CS1 AES-CBC-CS2 AES-CBC-CS3 AES-CFB1 AES-CFB128 AES-CFB8 AES-CTR AES-ECB AES-OFB AES-CCM AES-GCM AES-GMAC AES-CMAC HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512/224 HMAC-SHA2-512/256 HMAC-SHA3-224 HMAC-SHA3-256 HMAC-SHA3-384 HMAC-SHA3-512

Name	Type	Description	Properties	Algorithms
				KMAC-128 KMAC-256
KTS (AES KW, KWP)	BC-Auth		KTS:128, 192 or 256-bit keys provide between 128 and 256 bits of encryption strength	AES-KW AES-KWP
MAC AES (CMAC, GMAC)	MAC			AES-GMAC AES-CMAC
MAC HMAC	MAC			HMAC-SHA-1 HMAC-SHA2-224 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-512/224 HMAC-SHA2-512/256 HMAC-SHA3-224 HMAC-SHA3-256 HMAC-SHA3-384 HMAC-SHA3-512
MAC KMAC (XOF)	XOF			KMAC-128 KMAC-256
Message Digest	SHA			SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 SHA2-512/224 SHA2-512/256 SHA3-224 SHA3-256 SHA3-384 SHA3-512
Message Digest (XOF SHAKE)	XOF			SHAKE-128 SHAKE-256
Random	DRBG			Counter DRBG Hash DRBG HMAC DRBG
Signature DSA	DigSig-SigGen DigSig-SigVer			DSA SigGen (FIPS186-4) DSA SigVer (FIPS186-4)

Name	Type	Description	Properties	Algorithms
Signature ECDSA	DigSig-SigGen DigSig-SigVer			ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)
Signature EDDSA	DigSig-SigGen DigSig-SigVer			EDDSA SigGen EDDSA SigVer
Signature RSA	DigSig-SigGen DigSig-SigVer			RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-5) RSA Signature Primitive RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-5)

Table 16: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM:

The Module supports internal IV generation using the Approved DRBG. The IV is at least 96 bits in length per SP 800-38D Section 8.2.2, and the Approved DRBG generates outputs such that the (key, IV) pair collision probability is less than 2^{-32} per SP 800-38D Section 8.

AES-GCM IVs shall be used in compliance with FIPS 140-3 IG C.H scenario 1a (TLS/DTLS 1.2, per RFC 5288), 1d (SSHv2, per RFC 5647) and 5 (TLS 1.3, per RFC 8446). The Module is compatible with TLS/DTLS 1.2 protocol and provides the primitives to support the AES GCM ciphersuites from SP 800-52 Rev. 1 Section 3.3.1. The Module's implementation of AES-GCM is used together with one or more applications outside the Module's cryptographic boundary that implement the specified protocols; these protocols have not been reviewed or tested by the CAVP and CMVP. In each of the protocols, if the Module's power is lost and then restored, the key used for the AES GCM encryption/decryption shall be re-distributed. This condition is not enforced by the Module but is met implicitly. The Module does not retain any state across reset or power-cycles: AES-GCM key/IVs are not stored in non-volatile persistent memory (i.e., disk), hence no re-connection can occur without a fresh key establishment operation and the associated SSPs.

The Module explicitly ensures that the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values of 2^{64}-1 for a given session key. If this exhaustion condition is observed, the Module returns an error indication to the calling application, which will then need to either abort the connection, or trigger a handshake to establish a new encryption key.

XTS-AES:

In accordance with SP 800-38E, the XTS-AES algorithm is to be used for confidentiality on storage devices. The Module complies with FIPS 140-3 IG C.I by:

- Generating Key_1 and Key_2 independently according to the rules for component symmetric keys from SP 800-133 Rev. 2, Section 6.3.
- Explicitly checking that Key_1 ≠ Key_2 before using the keys in the XTS-AES algorithm to process data with them.

Key Agreement:

The Module implements the following Approved key agreement methods which have been CAVP tested and validated:

- KAS-ECC-SSC per SP 800-56A Rev. 3 (FIPS 140-3 IG D.F Scenario 2, path 1).
- KAS-FFC-SSC per SP 800-56A Rev. 3 (FIPS 140-3 IG D.F Scenario 2, path 1).
- KAS-IFC-SSC per SP 800-56B Rev. 2 (FIPS 140-3 IG D.F Scenario 1, path 1).

The Module obtains the FIPS 140-3 IG D.F required key agreement assurances:

- SP 800-56A Rev. 3 in accordance with Section 5.6.2.
- SP 800-56B Rev. 2 in accordance with Section 6.4.

PBKDF:

The implemented PBKDF uses Option 1a specified in SP 800-132 Section 5.4.

FIPS 140-3 IG D.N *SP 800-132 Password-Based Key Derivation for Storage Applications* notes that:

> *The strength of the Data Protection Key is based on the strength of the Password and/or Passphrase used in key derivation.* ***SP 800-132*** *does not impose any strictly defined requirements on the strength of a password. It says that "passwords* ***should*** *be strong enough so that it is infeasible for attackers to get access by guessing a password."*

The choice to use the PBKDF with a password or passphrase is entirely outside the scope of the Module, managed by the calling application – and potentially would need to accommodate not only application-level considerations, but end use environment considerations and policies as well. As examples, the end use environment may impose policies to reject words found in a dictionary, to use specific types of characters (upper case, lower case, punctuation) and so on. The Module does not enforce a reduced character space (referring to the set of allowed characters), and as such, any policy to restrict the character space weakens the potential strength of the derived Data Protection Key (KD_PW_PBKDF).

In the summary of password strength guidance below, the term *useful* refers to characters which are not simply padding the string, for example with some combination of repetitive characters – such means of skirting organizational policies are not recommended. The phrase *character space* refers to the set of characters that a password or passphrase is constrained to. The printable character space is assumed to be 95 printable characters.

Integrators making use of PBKDF with this Module shall determine password policy and input length based on the intended output key size and strength, taking into consideration the probability of guessing KD_PW_PBKDF. The following examples are provided to guide parameter selection:

- $1/(2^{256})$ = 8.6E-78 for a 32-byte KD_PW_PBKDF field with no character space restriction (equivalent to a 256-bit symmetric key).
- $1/(95^{18})$ = 2.5E-36 for KD_PW_PBKDF with 18 useful printable characters (better than a 112-bit symmetric key, i.e. $1/(2^{112})$ = 1.9E-34).
- $1/(95^{20})$ = 3.4E-48 for KD_PW_PBKDF with 20 useful printable characters (better than a 128-bit symmetric key, i.e. $1/(2^{128})$ = 2.9E-39).
- $1/(95^{40})$ = 7.8E-80 for KD_PW_PBKDF with 40 useful printable characters (better than a 256-bit symmetric key, i.e. $1/(2^{256})$ = 8.6E-78).

In accordance with SP 800-132 and FIPS 140-3 IG D.N, keys derived from passwords are only to be used in storage applications.

The iteration count shall be selected as large as possible, as long as the time required to generate the key using the entered password is acceptable for the users. The Module enforces the following SP 800-132 compliance checks:

- The iteration count is at least 1000.
- The salt length is at least 128 bits.
- The derived key length is at least 112 bits.

RSA:

The Module complies with FIPS 140-3 IG C.F as follows:

- RSA Key Generation, Signature Generation and Signature Verification have been tested and validated with all implemented modulus lengths for which CAVP testing is available: k = 1024 (legacy Signature Verification only), k = 2048, k = 3072, and k = 4096.
- The Module also supports RSA Key Generation, Signature Generation and Signature Verification with modulus lengths for which CAVP testing is not available: k > 4096.

SHA-3 and SHAKE:

The Module complies with FIPS 140-3 IG C.C as follows:

- All implemented SHA-3 and SHAKE functions have been tested and validated on all of the Module's operating environments.
- Vendor affirmation is claimed for use of the SHA3-256 and SHA3-512 hash functions as part of the Hash DRBG and HMAC DRBG, for which CAVP testing with SHA-3 is not available.

2.8 RBG and Entropy

N/A for this Module. The calling application is responsible for use of a SP 800-90B compliant entropy source outside the Module boundary providing at least 256 bits of security strength. Entropy is supplied to the Module via callback functions. The following caveat applies per FIPS 140-3 IG 9.3.A:

No assurance of the minimum strength of generated SSPs (e.g., keys).

2.9 Key Generation

The Module:

- Produces random values in accordance with SP 800-133 Rev. 2 Section 4, in that the DRBG output is provided directly as the random output.
- Does not provide any service beyond random value generation for symmetric key generation. SSPs used with symmetric key algorithms are provided by the calling application.
- Produces asymmetric keys in accordance with SP 800-133 Rev. 2 Section 5, in that all asymmetric keys generated by the Module (the Key management service) provide the output of the approved key generation algorithm with no post-processing or manipulation of the generated key pairs. As noted in the previous item, random values used in the asymmetric key generation algorithms are direct outputs of the DRBG. Keys produced by the Module use an internal Counter DRBG for which the minimum key size and equivalent security strength is 128 bits.
- Supports symmetric key derivation in accordance with SP 800-133 Rev. 2 Section 6.2, using the approved and CAVP listed KDF algorithms.

2.10 Key Establishment

The Module implements key agreement methods compliant with FIPS 140-3 IG D.F and key transport methods compliant with FIPS 140-3 IG D.G. Strengths are provided in Section 2.6.

2.11 Industry Protocols

The Module conforms to FIPS 140-3 IG D.C *References to the Support of Industry Protocols*: while it provides SP 800-56A Rev. 3 conformant schemes and API entry points oriented to TLS usage, the Module does not contain the full implementation of TLS. The following caveat is required:

No parts of the TLS protocol, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A (API - input)	Control Input Data Input	API input: stack frame including non-sensitive parameters.
N/A (API - output)	Data Output Status Output	API output: output parameters and return value resulting from call execution.

Table 17: Ports and Interfaces

The Module does not interact with physical ports. The Control Output interface is not applicable, as the Module does not control other components.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this Module.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
CO	Role	CO	

Table 18: Roles

The Module supports the mandatory Cryptographic Officer (CO) operational role only (implicitly identified), and does not support a maintenance role or a bypass capability. The Module does not provide an authentication or identification method of its own. The CO role is assumed by meeting the conditions of Section 11 of this document and in associated KeyPair FIPS Provider for OpenSSL 3 Guidance Documentation.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Cipher	Encrypt or decrypt data, including AEAD modes (CCM, GCM).	FIPS_OK	Encryption or decryption key; plaintext or ciphertext data; flags.	Status return. Plaintext or ciphertext data.	Cipher (Unauth) Cipher (Auth)	CO - SC_EDK_AES: W,E - SC_EDK_XTS: W,E
Get capabilities	Reports information on the requested capabilities.	FIPS_OK	Provider context, capability, callback pointer and arguments.	Description of capabilities.		
Initialize	Module initialization, including instantiation of the opaque (managed within the module) Counter DRBG instance.	FIPS_OK	Core handle, dispatch in and out, provider context.	Initialization status (1 = pass, 0 = fail).	Random MAC HMAC	CO - DRBG_EI: W,E,Z - DRBG_Seed: G,E,Z - DRBG_Key: G,W,E - DRBG_V: G,W,E
Key agreement	Perform key agreement primitives on behalf of the calling process (does not establish keys into the module).	FIPS_OK	Key structs (key agreement keys); flags.	Status return; key agreement shared secret.	CKG Section 5 Key agreement	CO - KAS_Private_ECC: W,E - KAS_Public_ECC: W,E - KAS_Private_FFC: W,E - KAS_Public_FFC: W,E - KAS_Private_IFC: W,E - KAS_Public_IFC: W,E - KAS_SS_ECC: G,R - KAS_SS_FFC: G,R - KAS_SS_IFC: G,R
Key derivation	Derive keying material from a shared secret.	FIPS_OK	Key agreement shared secret; flags.	Status return; derived keying material.	Key derivation CKG Section 6.2	CO - KD_DKM_KDF: G,R - KD_PW_PBKDF: W,E - KD_DKM_PBKDF: G,R - KD_SK: W,E
Key management	Generate asymmetric key pairs.	FIPS_OK	ECDSA, EdDSA: curve identifier. DSA, RSA: domain parameter targets.	Status return; general digital signature private and public keys.	Key management ECC Key management Edwards Key management FFC Key management IFC CKG Section 4	CO - DRBG_C: G,W,E - DRBG_Key: W,G,E - DRBG_V: W,G,E - GKP_Private_ECC: G,R - GKP_Public_ECC: G,R - GKP_Private_Edwards: G,R - GKP_Public_Edwards: G,R - GKP_Private_FFC: G,R - GKP_Public_FFC: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- GKP_Private_IFC: G,R - GKP_Public_IFC: G,R
Key transport	Encapsulate or decapsulate key material on behalf of the calling process.	FIPS_OK	Key encapsulation/decapsulation key or Key wrap/unwrap key.	Status return; key transport shared secret.	CKG Section 5 Key transport KTS (Cipher w/ CMAC, GMAC, HMAC, KMAC) KTS (AES KW, KWP)	CO - KTS_KDK_IFC: W,E - KTS_KEK_IFC: W,E - KTS_SS_IFC: G,R
Message authentication	Generate or verify data integrity.	FIPS_OK	Keyed hash key.	Status return; MAC output value.	MAC AES (CMAC, GMAC) MAC HMAC MAC KMAC (XOF)	CO - KH_Key_AES-CMAC: W,E - KH_Key_AES-GMAC: W,E - KH_Key_HMAC: W,E - KH_Key_KMAC: W,E
Message digest	Generate a message digest.	FIPS_OK	Message; flags.	Status return; Hash output value.	Message Digest Message Digest (XOF SHAKE)	
Query	Report available crypto operations.	FIPS_OK	Provider context, operation ID.	Array of available operations.		
Random	Generate random bits using the DRBG.	FIPS_OK	DRBG struct (RBG State); DRBG_Seed.	Status return; Random value.	Random CKG Section 4	CO - DRBG_C: W,E - DRBG_EI: W,E,Z - DRBG_Seed: G,E,Z - DRBG_Key: W,E - DRBG_V: W,E
Self-test	Perform the self-test sequence.	FIPS_OK	Provider context.	Status (1 = pass, 0 = fail).		
Show module name and versioning information	Return module name and versioning information.	FIPS_OK	Provider context, parameter types (array).	Parameter types (array) with: Name, Version.		
Show status	OpenSSL core metadata (Gettable parameters; Get parameters).	FIPS_OK	Provider context, parameter types (array).	Parameter types with: BuildInfo, Status, SecurityChecks; Status return.		
Signature	Generate or verify digital signatures. (SSPs are passed in by the calling process.)	FIPS_OK	Sign: signing key; message. Verify: signature value; flags; sizes.	Status return; Signature value.	CKG Section 5 Signature DSA Signature ECDSA Signature EDDSA Signature RSA	CO - DS_SGK_ECC: W,E - DS_SVK_ECC: W,E - DS_SGK_Edwards: W,E - DS_SVK_Edwards: W,E - DS_SGK_FFC: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DS_SVK_FFC: W,E - DS_SGK_IFC: W,E - DS_SVK_IFC: W,E
Teardown	Uninstantiate the module; zeroizes internal CTR DRBG state (DRBG_Key, DRBG_V).	FIPS_OK	Provider context.	None.		CO - DRBG_Key: Z - DRBG_V: Z
Zeroize	Zeroization of allocated key structures using openssl_cleanse.	FIPS_OK	Memory pointer.	Void.		CO - DRBG_C: Z - DRBG_EI: Z - DRBG_Key: Z - DRBG_Seed: Z - DRBG_V: Z - DS_SGK_ECC: Z - DS_SGK_Edwards: Z - DS_SGK_FFC: Z - DS_SGK_IFC: Z - DS_SVK_ECC: Z - DS_SVK_Edwards: Z - DS_SVK_FFC: Z - DS_SVK_IFC: Z - GKP_Private_ECC: Z - GKP_Private_Edwards: Z - GKP_Private_FFC: Z - GKP_Private_IFC: Z - GKP_Public_ECC: Z - GKP_Public_Edwards: Z - GKP_Public_FFC: Z - GKP_Public_IFC: Z - KAS_Private_ECC: Z - KAS_Private_FFC: Z - GKP_Private_ECC: Z - KAS_Private_IFC: Z - KAS_Public_ECC: Z - KAS_Public_FFC: Z - KAS_Public_IFC: Z - KAS_SS_ECC: Z - KD_DKM_KDF: Z - KD_DKM_PBKDF: Z - KD_SK: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- KH_Key_AES-CMAC: Z - KH_Key_AES-GMAC: Z - KH_Key_HMAC: Z - KH_Key_KMAC: Z - KTS_KDK_IFC: Z - KTS_KEK_IFC: Z - KTS_SS_IFC: Z - KAS_SS_ECC: Z - SC_EDK_AES: Z - SC_EDK_XTS: Z

Table 19: Approved Services

All services implemented by the Module correspond to the functionality described by the *fips_query* function, which returns available services based on an *operation_id* input.

The *fips_get_params* function provides access to the current status of the Module as well as the name and version; this information correlates to the validation listing. A 1 value returned in status indicates the Module is running without error (FIPS_OK); a 0 return indicates an error (with additional error details indicated as described in the release specific API documentation). Services are only operational in the running state. Any attempts to access services in any other state will result in an error being returned. If the integrity test or any CAST fails then any attempt to access any service will result in an error being returned.

The OpenSSL toolkit *OSSL_PROVIDER_get_params* function is used to invoke *fips_get_params*, when called with the Module's global handle and a pointer to a parameter structure (initialized using *provider_gettable_params* or the equivalent).

Regarding the Indicator of approved security services, the Module conforms to FIPS 140-3 IG 2.4.C *Approved Security Service Indicator*, similar to example 2. Each service provides context sensitive status responses as described in the OpenSSL 3 API manual pages; generally, functions of return type int return the value 1 for success with other error codes as appropriate for the call (described in API documentation).

The Module's name and version parameters (as cited in Section 2) along with the Module's internal indicators of the security-check and conditional-errors settings are used to confirm the Module is the validated Module operating in the approved mode with only approved security services.

Note that the caller provides the KAS_Private and KAS_Public keys for shared secret computation; the caller's exchange and assurance of PSPs with the remote participant is outside the scope of the Module.

4.4 Non-Approved Services

N/A for this Module.

4.5 External Software/Firmware Loaded

N/A for this Module.

5 Software/Firmware Security

5.1 Integrity Techniques

The Module uses HMAC-SHA2-256 as the approved integrity technique; the file fips.so.mac contains the integrity reference value. The Module is provided in an executable form (as fips.so shared object for use in Linux environments).

5.2 Initiate on Demand

The operator can initiate the integrity test on demand by calling *fips_self_test* (invoked using *OSSL_PROVIDER_self_test* called with the Module's global handle) or reloading the Module.

5.3 Open-Source Parameters

In accordance with ISO/IEC 19790:2012 Annex B, as the Module is open source, the tools used to build the Module as tested are:

- gcc version 9.3.0
- perl v5.30.0
- gnu make v4.2.1

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

No operational environment restrictions are required for operation in the approved mode. All conditions for operation of the Module in the approved mode are given in Section 2.4.

The Module conforms to FIPS 140-3 IG 2.3.C Processor Algorithm Accelerators (PAA) and Processor Algorithm Implementation (PAI). The AES-NI functions are identified by FIPS 140-3 IG 2.3.C as a known PAA.

7 Physical Security

N/A for this Module.

8 Non-Invasive Security

N/A for this Module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	R: Random access memory	Dynamic

Table 20: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
I	Calling process	Call stack (API) input parameters	Plaintext	Manual	Electronic	
O	Call stack (API) output parameters	Calling process	Plaintext	Manual	Electronic	

Table 21: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
C	C (Cleanse): Caller invocation of openssl_cleanse.	Overwrites with zeros	Caller invocation of openssl_cleanse
T	T (Teardown): Module unload - invokes cleanse internally.	Overwrites with zeros	Occurs when module is unloaded

Table 22: SSP Zeroization Methods

All SSPs are zeroized (overwritten with 0s) when they are no longer needed:

- CSPs and PSPs with a lifetime associated with an OpenSSL object (e.g., EVP_PKEY) are zeroized when freed or reinitialized. The *OPENSSL_cleanse* function is used to zeroize CSPs and PSPs owned by the caller.
- CSPs with a lifetime associated with the Module are zeroized on Module uninstantiation (the Teardown operation).

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG_C	Element of Hash DRBG state.	Size: 440-888 - Strength: 160 ≤ s ≤ 256	Hash_DRBG_C - CSP	Random		Random
DRBG_EI	Entropy input from an external source used for DRBG seeding.	Size: 128-2^35 - Strength: 128 ≤ s ≤ 256	Other - CSP			Random

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG_Key	Element of CTR DRBG or HMAC DRBG state.	Size: 128-256, 128-256 - Strength: 128 ≤ s ≤ 256, 160 ≤ s ≤ 256	CTR_DRBG_Key, HMAC_DRBG_Key - CSP	Random		Random
DRBG_Seed	Seed used for DRBG Instantiation and Reseed.	Size: 128-256 - Strength: 128 ≤ s ≤ 256	Other - CSP	Random		Random
DRBG_V	Element of CTR, Hash or HMAC DRBG state.	Size: 128-256, 128-256, 128-256 - Strength: 128 ≤ s ≤ 256, 128 ≤ s ≤ 256, 128 ≤ s ≤ 256	CTR_DRBG_Key, Hash_DRBG_Key, HMAC_DRBG_Key - CSP	Random		Random
DS_SGK_ECC	SigGen (private) key.	Size: 233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - Strength: s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 - CSP			Signature ECDSA
DS_SGK_Edwards	SigGen (private) key.	Size: 255, 448 - Strength: s = 128, s = 224	Edwards25519, Edwards448 - CSP			Signature EDDSA
DS_SGK_FFC	SigGen (private) key.	Size: 2048, 2048, 3072 - Strength: s = 112, s = 112, s = 128	L=2048/N=224, L=2048/N=256, L=3072/N=256 - CSP			Signature DSA
DS_SGK_IFC	SigGen (private) key.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, s = 128, s = 152, s = 176, s = 200	k=2048, k=3072, k=4096, k=6144, k=8192 - CSP			Signature RSA
DS_SVK_ECC	SigVer (public) key.	Size: 163, 233, 283, 409, 571, 163, 233, 283, 409, 571, 192, 224, 256, 384, 521 - Strength: s < 112, s = 112, s = 128, s = 192, s = 256, s < 112, s = 112, s = 128, s = 192, s = 256, s < 112, s = 112, s = 128, s = 192, s = 256	B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521 - PSP			Signature ECDSA
DS_SVK_Edwards	SigVer (public) key.	Size: 255, 448 - Strength: s = 128, s = 224	Edwards25519, Edwards448 - PSP			Signature EDDSA

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DS_SVK_FFC	SigVer (public) key.	Size: 1024, 2048, 2048, 3072 - Strength: s < 112, s = 112, s = 112, s = 128	L=1024/N=160, L=2048/N=224, L=2048/N=256, L=3072/N=256 - PSP			Signature DSA
DS_SVK_IFC	SigVer (public) key.	Size: 1024, 2048, 3072, 4096, 6144, 8192 - Strength: s ≤ 112, s = 112, s = 128, s = 152, s = 176, s = 200	k=1024, k=2048, k=3072, k=4096, k=6144, k=8192 - PSP			Signature RSA
GKP_Private_ECC	General ECDSA (private) key.	Size: 233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - Strength: s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 - CSP	Key management ECC		Key management ECC
GKP_Private_Edwards	General EdDSA (private) key.	Size: 255, 448 - Strength: s = 128, s = 224	Edwards25519, Edwards448 - CSP	Key management Edwards		Key management Edwards
GKP_Private_FFC	General FFC (private) key.	Size: 2048, 2048, 3072 - Strength: s = 112, s = 112, s = 128	L=2048/N=224, L=2048/N=256, L=3072/N=256 - CSP	Key management FFC		Key management FFC
GKP_Private_IFC	General RSA (private) key.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, s = 128, s = 152, s = 176, s = 200	k=2048, k=3072, k=4096, k=6144, k=8192 - CSP	Key management IFC		Key management IFC
GKP_Public_ECC	General ECDSA (public) key.	Size: 233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - Strength: s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 - PSP	Key management ECC		Key management ECC
GKP_Public_Edwards	General EdDSA (public) key.	Size: 255, 448 - Strength: s = 128, s = 224	Edwards25519, Edwards448 - PSP	Key management Edwards		Key management Edwards
GKP_Public_FFC	General FFC (public) key.	Size: 2048, 2048, 3072 - Strength: s = 112, s = 112, s = 128	L=2048/N=224, L=2048/N=256, L=3072/N=256 - PSP	Key management FFC		Key management FFC

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
GKP_Public_IFC	General RSA (public) key.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, s = 128, s = 152, s = 176, s = 200	k=2048, k=3072, k=4096, k=6144, k=8192 - PSP	Key management IFC		Key management IFC
KAS_Private_ECC	Key pair component used for shared secret generation.	Size: 233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - Strength: s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 - CSP			Key agreement
KAS_Private_FFC	Key pair component used for shared secret generation.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, 112 ≤ s ≤ 128, 112 ≤ s ≤ 152, 112 ≤ s ≤ 176, 112 ≤ s ≤ 200	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - CSP			Key agreement
KAS_Private_IFC	Key pair component used for shared secret generation.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, s = 128, s = 152, s = 176, s = 200	k=2048, k=3072, k=4096, k=6144, k=8192 - CSP			Key agreement
KAS_Public_ECC	Peer key pair component used for shared secret generation.	Size: 233, 283, 409, 571, 233, 283, 409, 571, 224, 256, 384, 521 - Strength: s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256, s = 112, s = 128, s = 192, s = 256	B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521 - PSP			Key agreement
KAS_Public_FFC	Peer key pair component used for shared secret generation.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, 112 ≤ s ≤ 128, 112 ≤ s ≤ 152, 112 ≤ s ≤ 176, 112 ≤ s ≤ 200	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 - PSP			Key agreement
KAS_Public_IFC	Peer key pair component used for shared secret generation.	Size: 2048, 3072, 4096, 6144, 8192 - Strength: s = 112, s = 128, s = 152, s = 176, s = 200	k=2048, k=3072, k=4096, k=6144, k=8192 - PSP			Key agreement

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KAS_SS_ECC	Shared secret calculation z output value (for KDF).	Size: 112 - 256 - Strength: 112 - 256	Other - CSP		Key agreement	Key agreement
KAS_SS_FFC	Shared secret calculation z output value (for KDF).	Size: 112 - 256 - Strength: 112 - 200	Other - CSP		Key agreement	Key agreement
KAS_SS_IFC	Shared secret calculation z output value (for KDF).	Size: 112 - 256 - Strength: 112 - 200	Other - CSP		Key agreement	Key agreement
KD_DKM_KDF	Key derivation derived keying material.	Size: 128 - 256 - Strength: 128 - 256	Other - CSP	Key derivation		Key derivation
KD_DKM_PBKDF	PBKDF derived key material	Size: 128 - Strength: 128	Other - CSP	Key derivation		Key derivation
KD_PW_PBKDF	PBKDF password input.	Size: 128 - Strength: 128	Other - CSP	Key derivation		Key derivation
KD_SK	Key derivation source key material.	Size: 128 - 256 - Strength: 128 - 256	Other - CSP			Key derivation
KH_Key_AES-CMAC	Keyed Hash key.	Size: 128, 192, 256 - Strength: s = 128, s = 192, s = 256	AES-128, AES-192, AES-256 - CSP			MAC AES (CMAC, GMAC)
KH_Key_AES-GMAC	Keyed Hash key.	Size: 128, 192, 256 - Strength: s = 128, s = 192, s = 256	AES-128, AES-192, AES-256 - CSP			MAC AES (CMAC, GMAC)
KH_Key_HMAC	Keyed Hash key.	Size: 112 - 2048 - Strength: 112 - 256	Other - CSP			MAC HMAC
KH_Key_KMAC	Keyed Hash key.	Size: 128, 256 - Strength: 112 ≤ s ≤ 128, 112 ≤ s ≤ 256	KMAC128, KMAC256 - CSP			MAC KMAC (XOF)
KTS_KDK_IFC	RSA key de-encapsulation Key (key transport).	Size: 2048, 3072, 4096, 6144 - Strength: s = 112, s = 128, s = 152, s = 176	Other - CSP			Key transport
KTS_KEK_IFC	RSA key encapsulation Key (key transport).	Size: 2048, 3072, 4096, 6144 - Strength: s = 112, s = 128, s = 152, s = 176	Other - PSP			Key transport
KTS_SS_IFC	RSA key transport shared secret.	Size: 112 - 256 - Strength: s = 112 - s = 176	Other - CSP		Key transport	Key transport

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SC_EDK_AES	Symmetric encryption and decryption.	Size: 128, 192, 256 - Strength: s = 128, s = 192, s = 256	AES-128, AES-192, AES-256 - CSP			Cipher (Unauth) Cipher (Auth)
SC_EDK_XTS	Symmetric encryption and decryption.	Size: 256, 512 - Strength: s = 128, s = 256	XTS-128, XTS-256 - CSP			Cipher (Unauth)

Table 23: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DRBG_C	I O	RAM:Plaintext	Call lifetime	C	DRBG_Seed:Derived From DRBG_V:Used with
DRBG_EI	I	RAM:Plaintext	Call lifetime	C	DRBG_Seed:Constituent
DRBG_Key	I O	RAM:Plaintext	Call lifetime (module up time for internal DRBG)	C T	DRBG_Seed:Derived From DRBG_V:Used with
DRBG_Seed		RAM:Plaintext	Call lifetime	C	DRBG_C:Derives DRBG_Key:Derives DRBG_V:Derives DRBG_EI:Incorporates
DRBG_V	I O	RAM:Plaintext	Call lifetime (module up time for internal DRBG)	C T	DRBG_Seed:Derived From DRBG_Key:Used with
DS_SGK_ECC	I	RAM:Plaintext	Call lifetime	C	DS_SVK_ECC:Paired With
DS_SGK_Edwards	I	RAM:Plaintext	Call lifetime	C	DS_SVK_Edwards:Paired With
DS_SGK_FFC	I	RAM:Plaintext	Call lifetime	C	DS_SVK_FFC:Paired With
DS_SGK_IFC	I	RAM:Plaintext	Call lifetime	C	DS_SVK_IFC:Paired With
DS_SVK_ECC	I	RAM:Plaintext	Call lifetime	C	DS_SGK_ECC:Paired With
DS_SVK_Edwards	I	RAM:Plaintext	Call lifetime	C	DS_SGK_Edwards:Paired With
DS_SVK_FFC	I	RAM:Plaintext	Call lifetime	C	DS_SGK_FFC:Paired With
DS_SVK_IFC	I	RAM:Plaintext	Call lifetime	C	DS_SGK_IFC:Paired With
GKP_Private_ECC	O	RAM:Plaintext	Call lifetime	C	GKP_Public_ECC:Paired With
GKP_Private_Edwards	O	RAM:Plaintext	Call lifetime	C	GKP_Public_Edwards:Paired With
GKP_Private_FFC	O	RAM:Plaintext	Call lifetime	C	GKP_Public_FFC:Paired With
GKP_Private_IFC	O	RAM:Plaintext	Call lifetime	C	GKP_Public_IFC:Paired With
GKP_Public_ECC	O	RAM:Plaintext	Call lifetime	C	GKP_Private_ECC:Paired With
GKP_Public_Edwards	O	RAM:Plaintext	Call lifetime	C	GKP_Private_Edwards:Paired With
GKP_Public_FFC	O	RAM:Plaintext	Call lifetime	C	GKP_Private_FFC:Paired With
GKP_Public_IFC	O	RAM:Plaintext	Call lifetime	C	GKP_Private_IFC:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KAS_Private_ECC	I	RAM:Plaintext	Call lifetime	C	KAS_Public_ECC:Paired With
KAS_Private_FFC	I	RAM:Plaintext	Call lifetime	C	KAS_Public_FFC:Paired With
KAS_Private_IFC	I	RAM:Plaintext	Call lifetime	C	KAS_Public_IFC:Paired With
KAS_Public_ECC	I	RAM:Plaintext	Call lifetime	C	KAS_Private_ECC:Paired With
KAS_Public_FFC	I	RAM:Plaintext	Call lifetime	C	KAS_Private_FFC:Paired With
KAS_Public_IFC	I	RAM:Plaintext	Call lifetime	C	KAS_Private_IFC:Paired With
KAS_SS_ECC	O	RAM:Plaintext	Call lifetime	C	KAS_Private_ECC:Calculated From KAS_Public_ECC:Calculated From
KAS_SS_FFC	O	RAM:Plaintext	Call lifetime	C	KAS_Private_FFC:Calculated From KAS_Public_FFC:Calculated From
KAS_SS_IFC	O	RAM:Plaintext	Call lifetime	C	KAS_Private_IFC:Calculated From KAS_Public_IFC:Calculated From
KD_DKM_KDF	O	RAM:Plaintext	Call lifetime	C	KD_SK:Derived From
KD_DKM_PBKDF	O	RAM:Plaintext	Call lifetime	C	KD_PW_PBKDF:Derived From
KD_PW_PBKDF	I	RAM:Plaintext	Call lifetime	C	KD_DKM_PBKDF:Derives
KD_SK	I	RAM:Plaintext	Call lifetime	C	KD_DKM_KDF:Derives
KH_Key_AES-CMAC	I	RAM:Plaintext	Call lifetime	C	
KH_Key_AES-GMAC	I	RAM:Plaintext	Call lifetime	C	
KH_Key_HMAC	I	RAM:Plaintext	Call lifetime	C	
KH_Key_KMAC	I	RAM:Plaintext	Call lifetime	C	
KTS_KDK_IFC	I	RAM:Plaintext	Call lifetime	C	KTS_SS_IFC:Unwraps
KTS_KEK_IFC	I	RAM:Plaintext	Call lifetime	C	KTS_SS_IFC:Wraps
KTS_SS_IFC	O	RAM:Plaintext	Call lifetime	C	KTS_KDK_IFC:Unwrapped By KTS_KEK_IFC:Wrapped By
SC_EDK_AES	I	RAM:Plaintext	Call lifetime	C	
SC_EDK_XTS	I	RAM:Plaintext	Call lifetime	C	

Table 24: SSP Table 2

Keys used for CASTs and the temporary value used in the integrity test are not SSPs; however, the latter is deleted after use as required by AS05.10.

The Module maintains only the Counter DRBG state used for key generation as a persistent CSP; this DRBG instance is used exclusively for approved services.

9.5 Additional Information

Key/Algorithm Type Equivalent Strengths: Reference sources for the strengths provided in SSP Table 1 are specified below. Equivalent strength is given for each key or algorithm type (as some algorithms do not use or produce keys).

Block Cipher (and related functions):

- AES (AES-128, AES-192, AES-256): SP 800-57 Part 1 Rev. 5 Table 2.

Digital Signature:

- ECC (B-163, B-233, B-283, B-409, B-571, K-163, K-233, K-283, K-409, K-571, P-192, P-224, P-256, P-384, P-521): SP 800-186 Table 1 (provides approximate elliptic curve security strengths). SP 800-186 and FIPS 140-3 IG C.K indicate that the Binary (B-) and Koblitz (K-) curves are deprecated.
- EdDSA (ED-25519, ED-448): SP 800-186 Table 1.
- FFC (DSA: L=1024/N=160, L=2048/N=224, L=2048/N=256, L=3072/N=256): SP 800-57 Part 1 Rev. 5 Table 2. Security strength for L=2048/N=256 is determined in accordance with FIPS 140-3 IG D.B Strength of SSP Establishment Methods as y = min(x, N/2), where x is 112 and therefore y = min(112, 128) = 112.
- IFC (RSA: k=1024, k=2048, k=3072, k=4096): SP 800-57 Part 1 Rev. 5 Table 2.

In Digital Signature applications, security strength is primarily associated with the asymmetric key pair specification. The hash function used must have equivalent strength equal to or greater than the security strength of the associated key pair.

Secure Hash (and related functions):

- SHA-1, SHA2 (SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA2-512/224, SHA2-512/256): SP 800-107 Rev. 1 Table 1.
- SHA3 (SHA3-224, SHA3-256, SHA3-384, SHA3-512): SP 800-57 Part 1 Rev. 5 Table 3.
- SHAKE (SHAKE128, SHAKE256): SP 800-185 Section 8.1.

Preimage resistance strength applies to hash algorithms used in DRBG, KDFs. Described also in SP 800-57 Part 1 Rev. 5 Table 3.

Message Authentication:

- KMAC (KMAC128, KMAC256): SP 800-56C Rev. 2 Table 3.

Key Agreement:

- KAS-ECC-SSC (B-233, B-283, B-409, B-571, K-233, K-283, K-409, K-571, P-224, P-256, P-384, P-521): SP 800-56A Rev. 3 Table 24.
- KAS-FFC-SSC (FB, FC, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, modp-2048, modp-3072, modp-4096, modp-6144, modp-8192): SP 800-56A Rev. 3 Tables 25 and 26.
- KAS-IFC-SSC (k=2048, k=3072, k=4096, k=6144, k=8192): SP 800-56B Rev. 2 Table 4 (provides approximate security strengths).

Key Agreement Key Derivation:

- KDA OneStep: SP 800-56C Rev. 2 Table 1 (hash), Table 2 (HMAC) and Table 3 (KMAC).

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
SW Integrity	HMAC-SHA2-256 #A4481	HMAC over the complete module file image	SW/FW Integrity	FIPS_OK or PROV_R_FIPS_MODULE_IN_ERROR_STATE	

Table 25: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB	128-bit	KAT	CAST	FIPS_OK	Encrypt	Performed on module load.
AES-ECB	128-bit	KAT	CAST	FIPS_OK	Decrypt	Performed on module load.
AES-GCM	256-bit	KAT	CAST	FIPS_OK	Encrypt	Performed on module load.
AES-GCM	256-bit	KAT	CAST	FIPS_OK	Decrypt	Performed on module load.
Counter DRBG	AES-128 with derivation function	KAT	CAST	FIPS_OK	Instantiate, Generate, Reseed	Performed on module load.
DSA SigGen (FIPS186-4)	2048-bit with SHA2-384	KAT	CAST	FIPS_OK	Sign	Performed on module load.
DSA SigVer (FIPS186-4)	2048-bit with SHA2-384	KAT	CAST	FIPS_OK	Verify	Performed on module load.
ECDSA SigGen (FIPS186-4)	P-224 with SHA2-512	KAT	CAST	FIPS_OK	Sign	Performed on module load.
ECDSA SigVer (FIPS186-4)	P-224 with SHA2-512	KAT	CAST	FIPS_OK	Verify	Performed on module load.
EDDSA ED448 SigGen	Edwards448 SigGen with SHA2-256	KAT	CAST	FIPS_OK	Sign	Performed on module load.
EDDSA ED448 SigVer	Edwards448 SigVer with SHA2-256	KAT	CAST	FIPS_OK	Verify	Performed on module load.
EDDSA ED25519 SigGen	Edwards25519 SigGen with SHA2-512	KAT	CAST	FIPS_OK	Sign	Performed on module load.
EDDSA ED25519 SigVer	Edwards25519 SigVer with SHA2-512	KAT	CAST	FIPS_OK	Verify	Performed on module load.
Hash DRBG	SHA2-256	KAT	CAST	FIPS_OK	Instantiate, Generate, Reseed	Performed on module load.
HMAC DRBG	SHA-1	KAT	CAST	FIPS_OK	Instantiate, Generate, Reseed	Performed on module load.
HMAC-SHA2-256	SHA2-256 with a 256-bit key	KAT	CAST	FIPS_OK	Generate	Performed on module load.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KAS-ECC-SSC Sp800-56Ar3	P-256	KAT	CAST	FIPS_OK	Ephemeral Unified Shared Secret (Z) Computation	Performed on module load.
KAS-FFC-SSC Sp800-56Ar3	L=2048/N=256	KAT	CAST	FIPS_OK	dhEphem Shared Secret (Z) Computation	Performed on module load.
KAS-IFC-SSC	k=2048	KAT	CAST	FIPS_OK	SP 800-56B Rev. 2 Section 8.2.2 RSA Primitive Computation	Performed on module load.
KAS-KDF OneStep SP800-56Cr2	SHA2-224	KAT	CAST	FIPS_OK	SP 800-56C Rev. 2 Section 4 OneStep KDF (AKA OpenSSL single-step or SS-KDF)	Performed on module load.
KAS-KDF TwoStep SP800-56Cr2	SHA2-256	KAT	CAST	FIPS_OK	SP 800-56C Rev. 2 Section 5 TwoStep KDF (HKDF variant)	Performed on module load.
KDF ANS 9.42	Fixed input KAT	KAT	CAST	FIPS_OK	SP 800-135 Rev. 1 Section 5.1 ANSI X9.42-2001 KDF KAT	Performed on module load.
KDF ANS 9.63	Fixed input KAT	KAT	CAST	FIPS_OK	SP 800-135 Rev. 1 Section 5.1 X9.63-2001 KDF KAT	Performed on module load.
KDF SP800-108	HMAC-SHA2-256	KAT	CAST	FIPS_OK	SP 800-108 Rev. 1 Section 4.1 KAT for a Counter Mode KDF	Performed on module load.
KDF SSH	Fixed input KAT	KAT	CAST	FIPS_OK	SP 800-135 Rev. 1 Section 5.2 SSHv2 KDF KAT	Performed on module load.
KTS-IFC	k=2048	KAT	CAST	FIPS_OK	SP 800-56B Rev. 2 Decrypt for CRT	Performed on module load.
KTS-IFC	k=2048	KAT	CAST	FIPS_OK	SP 800-56B Rev. 2 Encrypt for Basic	Performed on module load.
KTS-IFC	k=2048	KAT	CAST	FIPS_OK	SP 800-56B Rev. 2 Decrypt for Basic	Performed on module load.
PBKDF	SHA2-256, 24-byte password, 36-byte salt, iteration count of 4096	KAT	CAST	FIPS_OK	SP 800-132 Section 5.3 KAT of Master Key derivation	Performed on module load.
RSA SigGen (FIPS186-4)	k=2048 with SHA2-256	KAT	CAST	FIPS_OK	Sign	Performed on module load.
RSA SigVer (FIPS186-4)	k=2048 with SHA2-256	KAT	CAST	FIPS_OK	Verify	Performed on module load.
SHA-1	SHA-1	KAT	CAST	FIPS_OK	Simple SHA KAT	Performed on module load.
SHA2-512	SHA2-512	KAT	CAST	FIPS_OK	Simple SHA KAT	Performed on module load.
SHA3-256	SHA3-256	KAT	CAST	FIPS_OK	Simple SHA KAT	Performed on module load.
TLS v1.2 KDF RFC7627	Fixed input KAT	KAT	CAST	FIPS_OK	SP 800-135 Rev. 1 Section 4.2.2 TLS 1.2 KAT	Performed on module load.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
TLS v1.3 KDF	Fixed input KAT	KAT	CAST	FIPS_OK	RFC8446 Section 7.1 TLS v1.3 KDF KAT	Performed on module load.
DSA KeyGen (FIPS186-4)	PCT performed using the generated key pair	PCT	PCT	FIPS_OK	Sign, Verify	Performed on FFC (DSA, KAS-FFC-SSC) key pair generation, prior to returning the key pair on conclusion of the call.
ECDSA KeyGen (FIPS186-4)	PCT performed using the generated key pair	PCT	PCT	FIPS_OK	Sign, Verify	Performed on ECC (ECDSA) key pair generation, prior to returning the key pair on conclusion of the call.
EDDSA KeyGen	PCT performed using the generated key pair	PCT	PCT	FIPS_OK	Sign, Verify	Performed on Edwards (EdDSA) key pair generation, prior to returning the key pair on conclusion of the call.
RSA KeyGen (FIPS186-4)	PCT performed using the generated key pair	PCT	PCT	FIPS_OK	Sign, Verify	Performed on IFC (RSA, KAS-IFC-SSC, KTS-IFC) key pair generation, prior to returning the key pair on conclusion of the call.

Table 26: Conditional Self-Tests

The intended usage of asymmetric key pairs generated by the Module is not known at the time when the key pair is generated and the pairwise consistency test (PCT) is performed. In all cases, a sign and verify PCT is performed.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SW Integrity	HMAC over the complete module file image	SW/FW Integrity	On demand	Module load

Table 27: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-ECB	KAT	CAST	On demand	On power on or reset
AES-ECB	KAT	CAST	On demand	On power on or reset
AES-GCM	KAT	CAST	On demand	On power on or reset
AES-GCM	KAT	CAST	On demand	On power on or reset
Counter DRBG	KAT	CAST	On demand	On power on or reset
DSA SigGen (FIPS186-4)	KAT	CAST	On demand	On power on or reset
DSA SigVer (FIPS186-4)	KAT	CAST	On demand	On power on or reset
ECDSA SigGen (FIPS186-4)	KAT	CAST	On demand	On power on or reset
ECDSA SigVer (FIPS186-4)	KAT	CAST	On demand	On power on or reset
EDDSA ED448 SigGen	KAT	CAST	On demand	On power on or reset

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
EDDSA ED448 SigVer	KAT	CAST	On demand	On power on or reset
EDDSA ED25519 SigGen	KAT	CAST	On demand	On power on or reset
EDDSA ED25519 SigVer	KAT	CAST	On demand	On power on or reset
Hash DRBG	KAT	CAST	On demand	On power on or reset
HMAC DRBG	KAT	CAST	On demand	On power on or reset
HMAC-SHA2-256	KAT	CAST	On demand	On power on or reset
KAS-ECC-SSC Sp800-56Ar3	KAT	CAST	On demand	On power on or reset
KAS-FFC-SSC Sp800-56Ar3	KAT	CAST	On demand	On power on or reset
KAS-IFC-SSC	KAT	CAST	On demand	On power on or reset
KAS-KDF OneStep SP800-56Cr2	KAT	CAST	On demand	On power on or reset
KAS-KDF TwoStep SP800-56Cr2	KAT	CAST	On demand	On power on or reset
KDF ANS 9.42	KAT	CAST	On demand	On power on or reset
KDF ANS 9.63	KAT	CAST	On demand	On power on or reset
KDF SP800-108	KAT	CAST	On demand	On power on or reset
KDF SSH	KAT	CAST	On demand	On power on or reset
KTS-IFC	KAT	CAST	On demand	On power on or reset
KTS-IFC	KAT	CAST	On demand	On power on or reset
KTS-IFC	KAT	CAST	On demand	On power on or reset
PBKDF	KAT	CAST	On demand	On power on or reset
RSA SigGen (FIPS186-4)	KAT	CAST	On demand	On power on or reset
RSA SigVer (FIPS186-4)	KAT	CAST	On demand	On power on or reset
SHA-1	KAT	CAST	On demand	On power on or reset
SHA2-512	KAT	CAST	On demand	On power on or reset
SHA3-256	KAT	CAST	On demand	On power on or reset
TLS v1.2 KDF RFC7627	KAT	CAST	On demand	On power on or reset
TLS v1.3 KDF	KAT	CAST	On demand	On power on or reset
DSA KeyGen (FIPS186-4)	PCT	PCT	On demand	On power on or reset
ECDSA KeyGen (FIPS186-4)	PCT	PCT	On demand	On power on or reset
EDDSA KeyGen	PCT	PCT	On demand	On power on or reset
RSA KeyGen (FIPS186-4)	PCT	PCT	On demand	On power on or reset

Table 28: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Self-test failure	The self-test failure error state	If one of the KATs fails or integrity test fails	Reload the Module into memory	PROV_R_FIPS_MODULE_IN_ERROR_STATE

Table 29: Error States

10.5 Operator Initiation of Self-Tests

Each time the Module is powered up it tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged. The pre-operational self-tests are available on demand by reloading the Module.

On instantiation, the Module performs the pre-operational self-test and all CASTs. All KATs must complete successfully prior to any other use of cryptography by the Module.

The *fips_self_test* function (inclusive of software integrity verification) can also be called on demand, fulfilling AS05.11.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Module is based on the open-source OpenSSL 3 FIPS Provider, with a set of patches applied for conformance to FIPS 140-3 requirements.

The Module is provided to vendors who integrate it into their product, typically in a manufacturing environment, and is not provided directly to US or Canadian Federal agencies. Adherence to these steps maintains security throughout the distribution, build, installation and configuration processes. Tamper is detected via the use of HMAC integrity checking using a utility built with a separate FIPS 140-2 validated module.

An authorized Cryptographic Officer is required to perform these steps on each platform where it is intended to be used. The config file output contains information about the Module (such as the self-test status and the Module checksum) and must not be copied from one machine to another.
The Module is distributed by KeyPair to vendors as a tarball containing:

- The OpenSSL 3 source tarball and corresponding HMAC file (used to verify integrity);
- The KeyPair patch file and corresponding HMAC file (used to verify integrity);
- A shell script to apply the KeyPair patch;
- A shell script to perform the build and install process;
- A utility to verify integrity, called by the scripts;
- A demonstration application (and corresponding source and build script) that can be used to verify correct function, and to use as a sample application to understand how to use the Module.

The process to build and install the KeyPair Module is:

1) Untar (tar -xzvf) the distribution tarball.
2) Run the apply script.

   The apply script only needs to be run once, although reapplying patches is not harmful. This step includes:

   - the HMAC integrity check on the OpenSSL tarball;
   - the HMAC integrity check on the KeyPair patch set;
   - untar of the OpenSSL 3 tarball;
   - application of the patch set.
3) Run the make script

   The make script builds and installs all aspects of the OpenSSL 3 toolkit, including the FIPS provider with KeyPair patches for conformance to FIPS 140-3. The make script will:

   - Confirm install location;
   - Assure correct configuration for the build process;
   - Perform all build and install steps, including setting the configuration file as described in Section 2.4;
   - Print out status, as well as instructions to run the verify tool for assurance of the correct install and configuration of the Module.

11.2 Administrator Guidance

Additional Guidance Documentation inclusive of all information required per ISO/IEC 19790:2012 Section 7.11.9 is available from KeyPair Consulting by request.

11.3 Non-Administrator Guidance

N/A for this Module.

11.4 Design and Rules

The inherent properties of the Module are:

1. Manual key entry is not supported.
2. Data output is inhibited during self-tests, zeroization, SSP generation, and error states.
3. The Module does not perform any cryptographic function if any self-test has failed.

12 Mitigation of Other Attacks

12.1 Attack List

The Module implements mitigations for constant-time implementations and blinding attacks.

12.2 Mitigation Effectiveness

Constant-time implementations protect cryptographic implementations in the Module against timing analysis since such attacks exploit differences in execution time depending on the cryptographic operation, and constant-time implementations ensure that the variations in execution time cannot be traced back to the key, CSP or secret data.

Numeric blinding protects the RSA, DSA 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 blinding factor which is provided as an input to the decryption/signature operation and is discarded once the operation has completed and resulted in an output. This makes it difficult for attackers to attempt timing attacks on such operations without the knowledge of the blinding factor, and therefore the execution time cannot be correlated to the RSA/DSA/ECDSA key.

12.3 Guidance and Constraints

The mitigation mechanisms described in Section 12.2 are inherent within the validated algorithms. No other guidance or constraints are specified.