Apple Inc.

Apple corecrypto Module v13.0 [Apple silicon, User, Software, SL1]

FIPS 140-3 Non-Proprietary Security Policy

Prepared for:

Apple Inc.

One Apple Park Way

Cupertino, CA 95014

Prepared by:

atsec information security corporation

4516 Seton Center Parkway, Suite 250

Austin, TX 78759

www.atsec.com

Table of Contents

List of Tables

List of Figures

Trademarks

Apple's trademarks applicable to this document are listed in https://www.apple.com/legal/intellectual-property/trademark/appletmlist.html.

Other company, product, and service names may be trademarks or service marks of others.

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for Apple corecrypto Module v13.0 [Apple silicon, User, Software, SL1] cryptographic module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for a Security Level 1 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	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use: The Apple corecrypto Module v13.0 [Apple silicon, User, Software, SL1] cryptographic module (hereafter referred to as "the module") provides implementations of low-level cryptographic primitives to the Device OS's (iOS 16, iPadOS 16, watchOS 9, tvOS 16, T2OS 13 and macOS 13 Ventura) Security Framework and Common Crypto. The module provides services intended to protect data in transit and at rest.

The module is optimized for library use within the Device OS user space and does not contain any terminating assertions or exceptions. It is implemented as a Device OS dynamically loadable library. After the library is loaded, its cryptographic functions are made available to the Device OS application.

Any internal error detected by the module is returned to the caller with an appropriate return code. The calling Device OS application must examine the return code and act accordingly. The module communicates any error status synchronously through the use of its documented return codes, thus indicating the module's status. Caller-induced or internal errors do not reveal any sensitive material to callers.

Module Type: Software

Module Embodiment: MultiChipStand

Module Characteristics:

N/A

Cryptographic Boundary: The module cryptographic boundary is delineated by the dotted green rectangle in the Figure 1. The module executes within the user space of the computing platforms and operating systems listed in the Tested Operational Environments Table section 2.2.

Tested Operational Environment's Physical Perimeter (TOEPP): The physical perimeter is represented by the most exterior black line in the block diagram Figure 1.

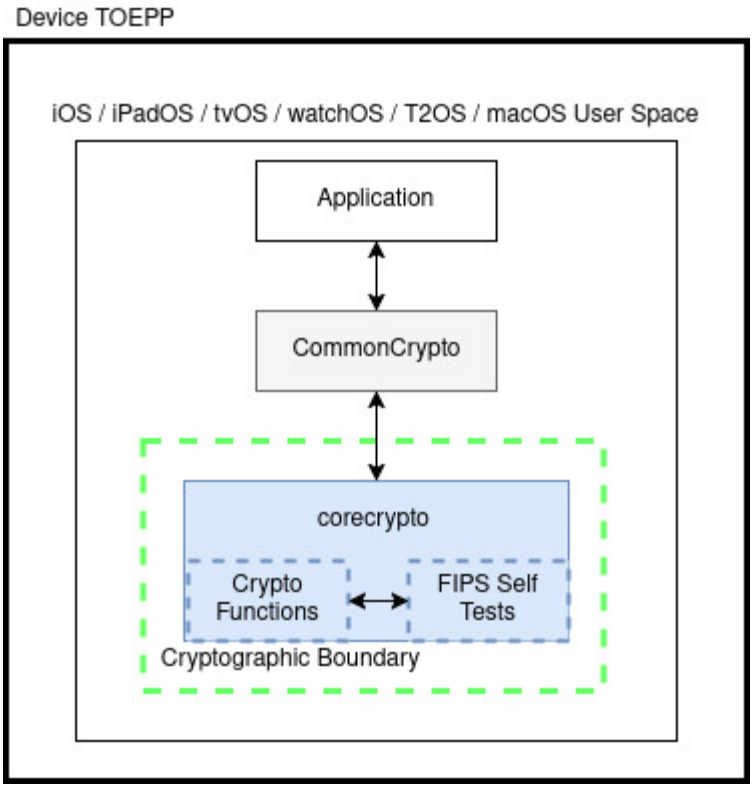

Figure 1: Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
corecrypto-1608.60.11	v13.0	N/A	HMAC-SHA-256

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

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
iPadOS 16	iPad (5th generation)	Apple A Series A9	Yes	NA	v13.0
iPadOS 16	iPad Pro 9.7-inch	Apple A Series A9X	Yes	NA	v13.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
iPadOS 16	iPad (7th generation)	Apple A Series A10 Fusion	Yes	NA	v13.0
iPadOS 16	iPad Pro 10.5-inch	Apple A Series A10X Fusion	Yes	NA	v13.0
iPadOS 16	iPad mini (5th generation)	Apple A Series A12 Bionic	Yes	NA	v13.0
iPadOS 16	iPad Pro 11-inch (1st generation)	Apple A Series A12X Bionic	Yes	NA	v13.0
iPadOS 16	iPad Pro 11-inch (2nd generation)	Apple A Series A12Z Bionic	Yes	NA	v13.0
iPadOS 16	iPad (9th generation)	Apple A Series A13 Bionic	Yes	NA	v13.0
iPadOS 16	iPad Air (4th generation)	Apple A Series A14 Bionic	Yes	NA	v13.0
iPadOS 16	iPad mini (6th generation)	Apple A Series A15 Bionic	Yes	NA	v13.0
iPadOS 16	iPad Pro 11-inch (3rd generation)	Apple M Series M1	Yes	NA	v13.0
iPadOS 16	iPad Pro 11-inch (4th generation)	Apple M Series M2	Yes	NA	v13.0
iOS 16	iPhone X	Apple A Series A11 Bionic	Yes	NA	v13.0
iOS 16	iPhone XS Max	Apple A Series A12 Bionic	Yes	NA	v13.0
iOS 16	iPhone 11 Pro	Apple A Series A13 Bionic	Yes	NA	v13.0
iOS 16	iPhone 12	Apple A Series A14 Bionic	Yes	NA	v13.0
iOS 16	iPhone 13 Pro Max	Apple A Series A15 Bionic	Yes	NA	v13.0
iOS 16	iPhone 14 Pro Max	Apple A Series A16 Bionic	Yes	NA	v13.0

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
watchOS 9	Apple Watch Series S4	Apple S Series S4	Yes	NA	v13.0
watchOS 9	Apple Watch Series S5	Apple S Series S5	Yes	NA	v13.0
watchOS 9	Apple Watch Series S6	Apple S Series S6	Yes	NA	v13.0
watchOS 9	Apple Watch Series S7	Apple S Series S7	Yes	NA	v13.0
watchOS 9	Apple Watch Series S8	Apple S Series S8	Yes	NA	v13.0
tvOS 16	Apple TV 4K	Apple A Series A10X Fusion	Yes	NA	v13.0
tvOS 16	Apple TV 4K (2nd generation)	Apple A Series A12 Bionic	Yes	NA	v13.0
tvOS 16	Apple TV 4K (3rd generation)	Apple A Series A15 Bionic	Yes	NA	v13.0
T2OS 13	Apple Security Chip T2	Apple T Series T2	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (13-inch, M1, 2020)	Apple M Series M1	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (16-inch, 2021)	Apple M Series M1 Pro	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (16-inch, 2021)	Apple M Series M1 Max	Yes	NA	v13.0
macOS 13 Ventura	Mac Studio	Apple M Series M1 Ultra	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (13-inch, M2, 2020)	Apple M Series M2	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (14-inch, 2023)	Apple M Series M2 Pro	Yes	NA	v13.0
macOS 13 Ventura	MacBook Pro (16-inch, 2023)	Apple M Series M2 Max	Yes	NA	v13.0

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

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

Operating System	Hardware Platform
iPadOS 16	iPad Pro 12.9-inch
iPadOS 16	iPad (6th generation)
iPadOS 16	iPad Pro 12.9-inch (2nd generation)
iPadOS 16	iPad Air (3rd generation)
iPadOS 16	iPad (8th generation)
iPadOS 16	iPad Pro 12.9-inch (3rd generation)
iPadOS 16	iPad Pro 12.9-inch (4th generation)
iPadOS 16	iPad Pro 12.9-inch (5th generation)
iPadOS 16	iPad Pro 12.9-inch (6th generation)
iOS 16	iPhone 8
iOS 16	iPhone 8 Plus
iOS 16	iPhone XS
iOS 16	iPhone XR
iOS 16	iPhone 11
iOS 16	iPhone 11 Pro Max
iOS 16	iPhone SE (2nd generation)
iOS 16	iPhone 12 mini
iOS 16	iPhone 12 Pro
iOS 16	iPhone 12 Pro Max
iOS 16	iPhone 13 mini
iOS 16	iPhone 13
iOS 16	iPhone 13 Pro
iOS 16	iPhone 14 Pro
watchOS 9	Apple Watch SE
macOS 13 Ventura	Mac mini

Operating System	Hardware Platform
macOS 13 Ventura	iMac (24-inch)
macOS 13 Ventura	MacBook Pro (14-inch, 2021)
macOS 13 Ventura	MacBook Air

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

None for this module

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved mode	Approved mode of operation is entered when the module utilizes the services that use the security functions listed in the Approved Algorithms Table and the Vendor Affirmed Algorithms Table.	Approved	The dedicated API function returns a '1' from fips_allowed_mode() for block cipher functions and fips_allowed() for all other services to indicate the executed cryptographic algorithm was approved
Non-Approved mode	Non-Approved mode of operation is entered when the module utilizes non-approved security functions in the Table Non-Approved Algorithms Not Allowed in the Approved Mode of Operation.	Non-Approved	The dedicated API function returns a '0' from fips_allowed_mode() for block cipher functions and fips_allowed() for all other services to indicate the executed cryptographic algorithm was non- approved

Table 5: Modes List and Description

2.5 Algorithms

Approved Algorithms:

AES-CBC

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3423	-	SP 800-38A
AES-CBC	A3424	-	SP 800-38A
AES-CBC	A3425	-	SP 800-38A
AES-CBC	A3426	-	SP 800-38A
AES-CBC	A3483	-	SP 800-38A
AES-CBC	A3484	-	SP 800-38A
AES-CBC	A3485	-	SP 800-38A
AES-CBC	A3486	-	SP 800-38A

Table 6: Approved Algorithms - AES-CBC

AES-CCM

Algorithm	CAVP Cert	Properties	Reference
AES-CCM	A3424	-	SP 800-38C
AES-CCM	A3426	-	SP 800-38C
AES-CCM	A3427	-	SP 800-38C
AES-CCM	A3484	-	SP 800-38C
AES-CCM	A3486	-	SP 800-38C
AES-CCM	A3487	-	SP 800-38C

Table 7: Approved Algorithms - AES-CCM

AES-CFB128

Algorithm	CAVP Cert	Properties	Reference
AES-CFB128	A3423	-	SP 800-38A
AES-CFB128	A3424	-	SP 800-38A
AES-CFB128	A3426	-	SP 800-38A
AES-CFB128	A3483	-	SP 800-38A
AES-CFB128	A3484	-	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-CFB128	A3486	-	SP 800-38A

Table 8: Approved Algorithms - AES-CFB128

AES-CFB8

Algorithm	CAVP Cert	Properties	Reference
AES-CFB8	A3424	-	SP 800-38A
AES-CFB8	A3426	-	SP 800-38A
AES-CFB8	A3484	-	SP 800-38A
AES-CFB8	A3486	-	SP 800-38A

Table 9: Approved Algorithms - AES-CFB8

AES-CMAC

Algorithm	CAVP Cert	Properties	Reference
AES-CMAC	A3426	-	SP 800-38B
AES-CMAC	A3486	-	SP 800-38B

Table 10: Approved Algorithms - AES-CMAC

AES-CTR

Algorithm	CAVP Cert	Properties	Reference
AES-CTR	A3424	-	SP 800-38A
AES-CTR	A3426	-	SP 800-38A
AES-CTR	A3427	-	SP 800-38A
AES-CTR	A3484	-	SP 800-38A
AES-CTR	A3486	-	SP 800-38A
AES-CTR	A3487	-	SP 800-38A

Table 11: Approved Algorithms - AES-CTR

AES-ECB

Algorithm	CAVP Cert	Properties	Reference
AES-ECB	A3423	-	SP 800-38A
AES-ECB	A3424	-	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-ECB	A3426	-	SP 800-38A
AES-ECB	A3427	-	SP 800-38A
AES-ECB	A3483	-	SP 800-38A
AES-ECB	A3484	-	SP 800-38A
AES-ECB	A3486	-	SP 800-38A
AES-ECB	A3487	-	SP 800-38A

Table 12: Approved Algorithms - AES-ECB

AES-GCM

Algorithm	CAVP Cert	Properties	Reference
AES-GCM	A3424	-	SP 800-38D
AES-GCM	A3426	-	SP 800-38D
AES-GCM	A3427	-	SP 800-38D
AES-GCM	A3484	-	SP 800-38D
AES-GCM	A3486	-	SP 800-38D
AES-GCM	A3487	-	SP 800-38D

Table 13: Approved Algorithms - AES-GCM

AES-KW

Algorithm	CAVP Cert	Properties	Reference
AES-KW	A3424	-	SP 800-38F
AES-KW	A3426	-	SP 800-38F
AES-KW	A3484	-	SP 800-38F
AES-KW	A3486	-	SP 800-38F

Table 14: Approved Algorithms - AES-KW

AES-OFB

Algorithm	CAVP Cert	Properties	Reference
AES-OFB	A3423	-	SP 800-38A
AES-OFB	A3424	-	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-OFB	A3426	-	SP 800-38A
AES-OFB	A3483	-	SP 800-38A
AES-OFB	A3484	-	SP 800-38A
AES-OFB	A3486	-	SP 800-38A

Table 15: Approved Algorithms - AES-OFB

AES-XTS

Algorithm	CAVP Cert	Properties	Reference
AES-XTS Testing Revision 2.0	A3423	-	SP 800-38E
AES-XTS Testing Revision 2.0	A3424	-	SP 800-38E
AES-XTS Testing Revision 2.0	A3426	-	SP 800-38E
AES-XTS Testing Revision 2.0	A3483	-	SP 800-38E
AES-XTS Testing Revision 2.0	A3484	-	SP 800-38E
AES-XTS Testing Revision 2.0	A3486	-	SP 800-38E

Table 16: Approved Algorithms - AES-XTS

CTR_DRBG

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A3424	-	SP 800-90A Rev. 1
Counter DRBG	A3426	-	SP 800-90A Rev. 1
Counter DRBG	A3427	-	SP 800-90A Rev. 1
Counter DRBG	A3484	-	SP 800-90A Rev. 1
Counter DRBG	A3486	-	SP 800-90A Rev. 1
Counter DRBG	A3487	-	SP 800-90A Rev. 1

Table 17: Approved Algorithms - CTR_DRBG

ECDSA-KEYGEN

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A3426	-	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A3428	-	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyGen (FIPS186-4)	A3486	-	FIPS 186-4
ECDSA KeyGen (FIPS186-4)	A3488	-	FIPS 186-4

Table 18: Approved Algorithms - ECDSA-KEYGEN

ECDSA-KEYVER

Algorithm	CAVP Cert	Properties	Reference
ECDSA KeyVer (FIPS186-4)	A3426	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3428	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3486	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3488	-	FIPS 186-4

Table 19: Approved Algorithms - ECDSA-KEYVER

ECDSA-SIGGEN

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigGen (FIPS186-4)	A3426	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3428	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3486	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3488	-	FIPS 186-4

Table 20: Approved Algorithms - ECDSA-SIGGEN

ECDSA-SIGVER

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-4)	A3426	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3428	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3486	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3488	-	FIPS 186-4

Table 21: Approved Algorithms - ECDSA-SIGVER

HMAC-SHA1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A3426	-	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A3428	-	FIPS 198-1
HMAC-SHA-1	A3486	-	FIPS 198-1
HMAC-SHA-1	A3488	-	FIPS 198-1

Table 22: Approved Algorithms - HMAC-SHA1

HMAC-SHA224

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-224	A3426	-	FIPS 198-1
HMAC-SHA2-224	A3428	-	FIPS 198-1
HMAC-SHA2-224	A3486	-	FIPS 198-1
HMAC-SHA2-224	A3488	-	FIPS 198-1

Table 23: Approved Algorithms - HMAC-SHA224

HMAC-SHA256

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-256	A3426	-	FIPS 198-1
HMAC-SHA2-256	A3428	-	FIPS 198-1
HMAC-SHA2-256	A3429	-	FIPS 198-1
HMAC-SHA2-256	A3486	-	FIPS 198-1
HMAC-SHA2-256	A3488	-	FIPS 198-1
HMAC-SHA2-256	A3489	-	FIPS 198-1

Table 24: Approved Algorithms - HMAC-SHA256

HMAC-SHA384

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-384	A3426	-	FIPS 198-1
HMAC-SHA2-384	A3428	-	FIPS 198-1
HMAC-SHA2-384	A3486	-	FIPS 198-1
HMAC-SHA2-384	A3488	-	FIPS 198-1

Table 25: Approved Algorithms - HMAC-SHA384

HMAC-SHA512

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512	A3426	-	FIPS 198-1
HMAC-SHA2-512	A3428	-	FIPS 198-1
HMAC-SHA2-512	A3486	-	FIPS 198-1
HMAC-SHA2-512	A3488	-	FIPS 198-1

Table 26: Approved Algorithms - HMAC-SHA512

HMAC-SHA512/256

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-512/256	A3426	-	FIPS 198-1
HMAC-SHA2-512/256	A3428	-	FIPS 198-1
HMAC-SHA2-512/256	A3486	-	FIPS 198-1
HMAC-SHA2-512/256	A3488	-	FIPS 198-1

Table 27: Approved Algorithms - HMAC-SHA512/256

KAS-ECC-SSC

Algorithm	CAVP Cert	Properties	Reference
KAS-ECC-SSC Sp800-56Ar3	A3426	-	SP 800-56A Rev. 3
KAS-ECC-SSC Sp800-56Ar3	A3486	-	SP 800-56A Rev. 3

Table 28: Approved Algorithms - KAS-ECC-SSC

KAS-FFC-SSC

Algorithm	CAVP Cert	Properties	Reference
KAS-FFC-SSC Sp800-56Ar3	A3426	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A3486	-	SP 800-56A Rev. 3

Table 29: Approved Algorithms - KAS-FFC-SSC

KBKDF

Algorithm	CAVP Cert	Properties	Reference
KDF SP800-108	A3426	-	SP 800-108 Rev. 1
KDF SP800-108	A3428	-	SP 800-108 Rev. 1

Algorithm	CAVP Cert	Properties	Reference
KDF SP800-108	A3486	-	SP 800-108 Rev. 1
KDF SP800-108	A3488	-	SP 800-108 Rev. 1

Table 30: Approved Algorithms - KBKDF

PBKDF

Algorithm	CAVP Cert	Properties	Reference
PBKDF	A3426	-	SP 800-132
PBKDF	A3428	-	SP 800-132
PBKDF	A3486	-	SP 800-132
PBKDF	A3488	-	SP 800-132

Table 31: Approved Algorithms - PBKDF

RSA-KEYGEN

Algorithm	CAVP Cert	Properties	Reference
RSA KeyGen (FIPS186-4)	A3426	-	FIPS 186-4
RSA KeyGen (FIPS186-4)	A3428	-	FIPS 186-4
RSA KeyGen (FIPS186-4)	A3486	-	FIPS 186-4
RSA KeyGen (FIPS186-4)	A3488	-	FIPS 186-4

Table 32: Approved Algorithms - RSA-KEYGEN

RSA-SIGGEN

Algorithm	CAVP Cert	Properties	Reference
RSA SigGen (FIPS186-4)	A3426	-	FIPS 186-4
RSA SigGen (FIPS186-4)	A3428	-	FIPS 186-4
RSA SigGen (FIPS186-4)	A3486	-	FIPS 186-4
RSA SigGen (FIPS186-4)	A3488	-	FIPS 186-4

Table 33: Approved Algorithms - RSA-SIGGEN

RSA-SIGVER

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-4)	A3426	-	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
RSA SigVer (FIPS186-4)	A3428	-	FIPS 186-4
RSA SigVer (FIPS186-4)	A3486	-	FIPS 186-4
RSA SigVer (FIPS186-4)	A3488	-	FIPS 186-4

Table 34: Approved Algorithms - RSA-SIGVER

SAFEPRIME-KEYGEN

Algorithm	CAVP Cert	Properties	Reference
Safe Primes Key Generation	A3426	-	SP 800-56A Rev. 3
Safe Primes Key Generation	A3486	-	SP 800-56A Rev. 3

Table 35: Approved Algorithms - SAFEPRIME-KEYGEN

SHA1

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A3426	-	FIPS 180-4
SHA-1	A3428	-	FIPS 180-4
SHA-1	A3486	-	FIPS 180-4
SHA-1	A3488	-	FIPS 180-4

Table 36: Approved Algorithms - SHA1

SHA224

Algorithm	CAVP Cert	Properties	Reference
SHA2-224	A3426	-	FIPS 180-4
SHA2-224	A3428	-	FIPS 180-4
SHA2-224	A3486	-	FIPS 180-4
SHA2-224	A3488	-	FIPS 180-4

Table 37: Approved Algorithms - SHA224

SHA256

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A3426	-	FIPS 180-4
SHA2-256	A3428	-	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A3429	-	FIPS 180-4
SHA2-256	A3486	-	FIPS 180-4
SHA2-256	A3488	-	FIPS 180-4
SHA2-256	A3489	-	FIPS 180-4

Table 38: Approved Algorithms - SHA256

SHA384

Algorithm	CAVP Cert	Properties	Reference
SHA2-384	A3426	-	FIPS 180-4
SHA2-384	A3428	-	FIPS 180-4
SHA2-384	A3486	-	FIPS 180-4
SHA2-384	A3488	-	FIPS 180-4

Table 39: Approved Algorithms - SHA384

SHA512

Algorithm	CAVP Cert	Properties	Reference
SHA2-512	A3426	-	FIPS 180-4
SHA2-512	A3428	-	FIPS 180-4
SHA2-512	A3486	-	FIPS 180-4
SHA2-512	A3488	-	FIPS 180-4

Table 40: Approved Algorithms - SHA512

SHA512/256

Algorithm	CAVP Cert	Properties	Reference
SHA2-512/256	A3426	-	FIPS 180-4
SHA2-512/256	A3428	-	FIPS 180-4
SHA2-512/256	A3486	-	FIPS 180-4
SHA2-512/256	A3488	-	FIPS 180-4

Table 41: Approved Algorithms - SHA512/256

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
Cryptographic Key Generation (CKG)	RSA Key Generation:Modulus: 2048, 3072, 4096; Key strength: from 112 to 150-bits ECDSA Key Generation:P-224, P-256, P-384, P-521; Key strength: from 112 to 256-bits Safe Prime Key Generation:Safe prime groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Key strength: from 112 to 200-bits	Apple corecrypto Module [Apple ARM, User, Software, SL1] (c_ltc)	FIPS 140-3 IG D.H. and [SP800-133rev2] sections 4 and 5.1
CKG	RSA Key Generation:Modulus: 2048, 3072, 4096; Key strength: from 112 to 150-bits ECDSA Key Generation:P-224, P-256, P-384, P-521; Key strength: from 112 to 256-bits Safe Prime Key Generation:MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Key strength: from 112 to 200-bits	Apple corecrypto Module [Apple ARM, User, Software, SL1] (vng_ltc)	FIPS 140-3 IG D.H. and [SP800-133rev2] sections 4 and 5.1

Table 42: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

There are no non-Approved but "Allowed functions" with security claimed algorithms in approved mode.

Non-Approved, Allowed Algorithms with No Security Claimed:

Name	Caveat	Use and Function
MD5	Allowed in Approved mode with no security claimed per IG 2.4.A Digest Size: 128-bit	Message Digest (used as part of the TLS key establishment scheme v1.0, v1.1 only)

Table 43: Non-Approved, Allowed Algorithms with No Security Claimed

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
ANSI X9.63 KDF	Hash based Key Derivation Function

Name	Use and Function
Blowfish	Encryption / Decryption
CAST5	Encryption / Decryption Key Sizes: 40 to 128 bits in 8-bit increments
DES	Encryption / Decryption Key Size: 56-bits
Diffie-Hellman	Shared Secret Computation using key size < 2048
ECDSA	Generation / Verification / SigGen / SigVer with curve P-192
ECDSA KeyGen	Key Pair Generation for compact point representation of points
EC Diffie-Hellman	Shared Secret Computation using curves < P-224
EdDSA	Key Generation, Signature Generation, Signature Verification with Ed25519
ECDH	Key agreement with X25519
HKDF [SP800-56Crev2]	Key Derivation Function
Integrated Encryption Scheme on elliptic curves (ECIES)	Hybrid encryption scheme
MD2	Message Digest size: 128-bit
MD4	Message Digest size: 128-bit
MD5 (except in the TLS 1.0/1.1 context)	Message Digest size: 128-bit
OMAC (One-Key CBC MAC)	MAC generation / verification
RC2	Encryption / Decryption Key Sizes 8 to 1024-bits
RC4	Encryption / Decryption Key Sizes 8 to 4096-bits
RFC6637	Key Derivation Function
RIPEMD	Message Digest size: 160-bits
RSA KeyGen	ANSI X9.31 Key Pair Generation with Key Size < 2048
RSA SigGen	PKCS#1 v1.5 and PSS; Signature Generation Key Size < 2048
RSA SigVer	Signature Verification Key Size < 1024
RSA Key Wrapping	OAEP, PKCS#1 v1.5 and -PSS schemes

Name	Use and Function
Triple-DES [SP 800-67r2]	CBC, CTR, CFB64, ECB, CFB8, OFB
SHA-3	Message Digest
HPKE (Hybrid Public Key Encryption) [RFC9180]	Hybrid encryption scheme
Keccak	Message Digest

Table 44: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Unauthenticated Symmetric Encryption and Decryption	BC-UnAuth	Key Size / Key Strength: 128, 192, 256-bits (for all but XTS, which supports 128 and 256 bit keys)	AES [FIPS 197; SP 800-38A]:CBC AES [FIPS 197; SP 800-38A]:CFB128 AES [FIPS 197; SP 800-38A]:CFB8 AES [FIPS 197; SP 800-38A]:CTR AES [FIPS 197; SP 800-38A]:ECB AES [FIPS 197; SP 800-38A]:OFB AES [FIPS 197; SP 800-38E]:XTS	AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CBC AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB128 AES-CFB8 AES-CFB8 AES-CFB8 AES-CFB8 AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-CTR AES-ECB AES-ECB AES-ECB AES-ECB AES-ECB

Name	Type	Description	Properties	Algorithms
				AES-ECB AES-ECB AES-ECB 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
Authenticated Symmetric Encryption and Decryption	BC-Auth	Key Size/ Key Strength: 128, 192, 256-bits	AES [FIPS 197; SP 800-38C]:CCM AES [FIPS 197; SP 800-38D]:GCM	AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-CCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM AES-GCM

Name	Type	Description	Properties	Algorithms
Random Number Generation	DRBG	Key Size/ Key Strength: 128, 256- bits. Derivation Function Enabled, No Prediction Resistance	DRBG [SP800-90ARev1]:CTR_DRBG; AES-128, AES-256	Counter DRBG Counter DRBG Counter DRBG Counter DRBG Counter DRBG Counter DRBG
ECDSA Asymmetric Key Generation	AsymKeyPair-KeyGen CKG	Curve: P-224, P-256, P-384, P-521. Key Strength: from 112 to 256-bits	ECDSA ANSI X9.62 [FIPS 186-4]:Key Pair Generation (CKG using method in Sections 4 and 5.1 [SP 800-133Rev2]). Testing Candidates	ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4) ECDSA KeyGen (FIPS186-4)
ECDSA Asymmetric Key Verification	AsymKeyPair-KeyVer	Curve: P-224, P-256, P-384, P-521. Key Strength: from 112 to 256-bits	ECDSA ANSI X9.62 [FIPS 186-4]:N/A	ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4) ECDSA KeyVer (FIPS186-4)
ECDSA Digital Signature Generation	DigSig-SigGen	Curve: P-224, P-256, P-384, P-521. Key Strength: from 112 to 256-bits	ECDSA ANSI X9.62 [FIPS 186-4]:SHA2-224, SHA2-256, SHA2- 384, SHA2-512	ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4) ECDSA SigGen (FIPS186-4)
ECDSA Digital Signature Verification	DigSig-SigVer	Curve: P-224, P-256, P-384, P-521. Key Strength: from 112 to 256-bits	ECDSA ANSI X9.62 [FIPS 186-4]:SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4) ECDSA SigVer (FIPS186-4)

Name	Type	Description	Properties	Algorithms
				ECDSA SigVer (FIPS186-4)
CMAC Message Authentication	MAC	Key Size/ Key Strength: 128, 192, 256-bits	AES [FIPS 197; SP 800-38B]:CMAC	AES-CMAC AES-CMAC
HMAC Message Authentication	MAC	Key Size: 8 - 262144 bits	HMAC SHA1 [FIPS 198]:Key Strength: 128 HMAC SHA224 [FIPS 198]:Key Strength: 224 HMAC SHA256 [FIPS 198]:Key Strength: 256 HMAC SHA384 [FIPS 198]:Key Strength: 384 HMAC SHA512 [FIPS 198]:Key Strength: 512 HMAC SHA512/256 [FIPS 198]:Key Strength: 256	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-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-384 HMAC-SHA2-512 HMAC-SHA2-

Name	Type	Description	Properties	Algorithms
				512 HMAC-SHA2-512 HMAC-SHA2-512 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256 HMAC-SHA2-512/256
ECC Shared Secret Computation	KAS-SSC	P-224, P-256, P-384, P-521. Key Strength: from 112 to 256-bits	KAS-ECC-SSC [SP800-56ARev 3] and FIPS 140-3 IG D.F scenario 2 path 1:Scheme: ephemeral Unified KAS Role: initiator, responder	KAS-ECC-SSC Sp800-56Ar3 KAS-ECC-SSC Sp800-56Ar3
FFC Shared Secret Computation	KAS-SSC	MODP-2048, MODP-3072, MODP- 4096, MODP-6144, MODP-8192. Key Strength: from 112 to 200	KAS-FFC-SSC [SP800- 56ARev3] and FIPS 140-3 IG D.F scenario 2 path 1:Scheme: dh Ephem with safe prime groups KAS Role: initiator, responder	KAS-FFC-SSC Sp800-56Ar3 KAS-FFC-SSC Sp800-56Ar3
KBKDF Key Derivation with HMAC	KBKDF	Key Size / Key Strength: 128 - 256 bits Supported [output] Lengths: 8-4096 Increment 8 Fixed Data Order: Before Fixed Data	KBKDF [SP800-108r1]:KDF Mode: Counter and Feedback MAC Mode: HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2- 256, HMAC-SHA2-384, HMAC- SHA2-512; Counter Length: 32	KDF SP800-108 KDF SP800-108 KDF SP800-108 KDF SP800-108

Name	Type	Description	Properties	Algorithms
Key wrapping/ Key unwrapping	KTS-Wrap	Key Size/ Key Strength: 128, 192, 256-bits	KTS (AES) [FIPS 197; SP 800-38 F]:AES-KW	AES-KW AES-KW AES-KW AES-KW
RSA Asymmetric Key Generation	AsymKeyPair-KeyGen CKG	Key Size: 2048, 3072, 4096-bits. Key Strength: from 112 to 150-bits	RSA [FIPS 186- 4]; ANSI X9.31:CKG using method in Sections 4 and 5.1 [SP 800-133Rev2]	RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4) RSA KeyGen (FIPS186-4)
RSA Digital Signature Generation	DigSig-SigGen	Key Size: 2048, 3072, 4096-bits. Key Strength: from 112 to 150-bits	RSA [FIPS 186-4]:PKCS#1 v1.5 and PKCS PSS	RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4) RSA SigGen (FIPS186-4)
RSA Digital Signature Verification	DigSig-SigVer	Key Size: 1024, 2048, 3072, 4096- bits. Key Strength: from 80 to 150- bits	RSA [FIPS 186-4]:PKCS#1 v1.5 and PKCS PSS	RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4) RSA SigVer (FIPS186-4)
Safeprime Key Generation	AsymKeyPair-KeyGen CKG	Safe Prime Groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192. Key Strength: from 112 to 200-bits	Safe Primes Key Generation:Safe Prime Groups: MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192; CKG using method in Sections 4 and 5.1 [SP 800-133Rev2]	Safe Primes Key Generation Safe Primes Key Generation

Name	Type	Description	Properties	Algorithms
Message Digest	SHA	N/A	SHS [FIPS 180-4]:SHA1 SHS [FIPS 180-4]:SHA224 SHS [FIPS 180-4]:SHA256 SHS [FIPS 180-4]:SHA384 SHS [FIPS 180-4]:SHA512 SHS [FIPS 180-4]:SHA512/256	SHA-1 SHA-1 SHA-1 SHA-1 SHA2-224 SHA2-224 SHA2-224 SHA2-224 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-256 SHA2-384 SHA2-384 SHA2-384 SHA2-384 SHA2-512 SHA2-512 SHA2-512 SHA2-512 SHA2-512/256 SHA2-512/256 SHA2-512/256 SHA2-512/256
PBKDF Key Derivation	PBKDF	Key Size / Key Strength: 128 - 256 bits Password length: 8- 128 bytes Increment 1 Salt Length: 128-4096 Increment 8 Iteration Count: 10-1000 Increment 1	PBKDF [SP800-132]:HMAC with: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	PBKDF PBKDF PBKDF PBKDF

Name	Type	Description	Properties	Algorithms
KBKDF Key Derivation with CMAC	KBKDF	Key Size / Key Strength: 128 - 256 bits Supported [output] Lengths: 8-4096 Increment 8 Fixed Data Order: Before Fixed Data Counter Length: 8, 16, 24, 32	KBKDF [SP800-108r1]:KDF Mode: Counter CMAC Mode: CMAC-AES128, CMAC-AES192, CMAC-AES256	KDF SP800-108 KDF SP800-108

Table 45: Security Function Implementations

2.7 Algorithm Specific Information

GCM IV

AES-GCM IV is constructed in compliance with IG C.H scenario 1 (TLS 1.2) and scenario 2 (IPsec-v3). Users should consult IG C.H specific scenario for all the details and requirements of using AES-GCM mode.

The GCM IV generation follows RFC 5288 and shall only be used for the TLS protocol version 1.2. The counter portion of the IV is set by the module within its cryptographic boundary. 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 nonce_explicit, or counter portion of the IV will not exhaust all of its possible values.

The GCM IV generation follows RFC 4106 and shall only be used for the IPsec-v3 protocol version 3. The counter portion of the IV is set by the module within its cryptographic boundary. The module does not implement the IPsec 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 IPsec protocol implicitly ensures that the nonce_explicit, or counter portion of the IV will not exhaust all of its possible values.

In both protocols in case 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; however, it is met implicitly. The module does not retain any state when power is lost. As indicated in Table 11, column Storage, the module exclusively uses volatile storage. This means that AES-GCM key/IVs are not persistently stored during power off: therefore, there is no re-connection possible when the power is back on with re-generation of the key used for GCM. After restoration of the power, the user of the module (e.g., TLS, IKE) along with User application

that implements the protocol, must perform a complete new key establishment operation using new random numbers (Entropy input string, DRBG seed, DRBG internal state V and Key, shared secret values that are not retained during power cycle, see table 11) with subsequent KDF operations to establish a new GCM key/IV pair on either side of the network communication channel.

AES-XTS

AES-XTS mode is only approved for hardware storage applications. The length of the AES-XTS data unit does not exceed 2^{20} blocks. The module checks explicitly that Key_1 ≠ Key_2 before using the keys in the XTS-Algorithm to process data with them compliant with IG C.I.

Key Derivation using SP 800-132 PBKDF2

The module implements a CAVP compliance tested key derivation function compliant to [SP800-132]. The service returns the key derived from the provided password to the caller. The length of the password used as input to PBKDFv2 shall be at least 8 characters and the worst-case probability of guessing the value is 10^8 assuming all characters are digits only. The user shall choose the password length and the iteration count in such a way that the combination will make the key derivation computationally intensive. PBKDFv2 is implemented to support the option 1a specified in section 5.4 of [SP800-132]. The keys derived from [SP800-132] map to section 4.1 of [SP800-133rev2] as indirect generation from DRBG. The derived keys may only be used in storage applications.

2.8 RBG and Entropy

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Apple corecrypto physical entropy source	Physical	See Tested Operational Environment Table in section 2.2	256 bit	256 bit	SHA-256 [ACVP cert. # C1223]
Apple corecrypto non- physical entropy source	Non-Physical	See Tested Operational Environment Table in section 2.2	256 bit	256 bit	SHA-256 [ACVP Certs. # A3687, A3522]

Table 46: Entropy Sources

Entropy sources: Two entropy sources (one non-physical entropy source and one physical entropy source) residing within the TOEPP provide the random bits. The entropy sources are located within the physical perimeter of the module (TOEPP) but outside the cryptographic boundary of the module.

RBGs: The NIST [SP 800-90ARev1] approved deterministic random bit generators (DRBG) used for random number generation is a CTR_DRBG using AES-256 with derivation function and without prediction resistance.

The module performs DRBG health tests according to [SP800-90ARev1 section 11.3].

The deterministic random bit generators are seeded by /dev/random. The /dev/random is the User Space interface.

RBG Output: The output of entropy sources provides 256-bits of entropy to seed and reseed SP800-90ARev1 DRBG during initialization (seed) and reseeding (reseed).

2.9 Key Generation

The module generates RSA, Diffie-Hellman, and ECDSA and EC Diffie-Hellman keys and SSPs in accordance with FIPS 140-3 IG D.H. The cryptographic module performs Cryptographic Key Generation (CKG) for asymmetric keys as per sections 4 and 5.1 [SP800-133r2] (vendor affirmed), compliant with [FIPS186-4], and using DRBG compliant with [SP800-90ARev1]. A seed (i.e., the random value) used in asymmetric key generation is a direct output from [SP800-90ARev1] CTR_DRBG. The key generation service for RSA, Diffie-Hellman, ECDSA and EC Diffie-Hellman key pairs as well as the [SP 800-90ARev1] DRBG have been ACVT tested with algorithm certificates found in the Approved Algorithms Table.

2.10 Key Establishment

The module provides the following key/SSP establishment services in the Approved mode:

- PBKDF Key Derivation

The module implements a CAVP compliance tested key derivation function compliant to [SP800-132]. The service returns the key derived from the provided password to the caller. The length of the password used as input to PBKDFv2 shall be at least 8 characters and the worst-case probability of guessing the value is 10^8 assuming all characters are digits only. The user shall choose the password length and the iteration count in such a way that the combination will make the key derivation computationally intensive. PBKDFv2 is implemented to support the option 1a specified in section 5.4 of [SP800-132]. The keys derived from [SP800-132] map to section 4.1 of [SP800-133Rev2] as indirect generation from DRBG. The derived keys may only be used in storage applications.

- KBKDF Key Derivation

The KBKDF is compliant to [SP800-108Rev1]. The module implements both Counter and Feedback modes with HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, or HMAC-SHA2-512 as the PRF.

Key Agreement Information: See - Security Function Implementations Table, line KAS and details below:

- Diffie-Hellman Shared Secret Computation

The module provides SP800-56ARev3 compliant key establishment according to FIPS 140-3 IG D.F scenario 2 path (1) with Diffie-Hellman shared secret computation. The shared secret computation provides between 112 and 200 bits of encryption strength.

- EC Diffie-Hellman Shared Secret Computation

The module provides SP800-56ARev3 compliant key establishment according to FIPS 140-3 IG D.F scenario 2 path (1) with EC Diffie-Hellman shared secret computation. The shared secret computation provides between 112 and 256 bits of encryption strength.

Key Transport Information: See - Security Function Implementations Table, line KTS and details below:

- AES-Key Wrapping

The module implements a Key Transport Scheme (KTS) using AES-KW compliant to [SP800-38F]. The SSP establishment methodology provides between 128 and 256 bits of encryption strength.

2.11 Industry Protocols

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

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input Data Output	Data inputs/outputs are provided in the variables passed in the C language Application Programming Interfaces (APIs) and callable service invocations, generally through caller-supplied buffers
N/A	Control Input	Control inputs which control the mode of the module are provided through dedicated parameters.
N/A	Status Output	Status output is provided in return codes and through messages. Documentation for each API lists possible return codes. A complete list of all return codes returned by the C language APIs within the module is provided in the header files and the API documentation. Messages are also documented in the API documentation.

Table 47: Ports and Interfaces

The module does not implement a Control Output Logical Interface

4 Roles, Services, and Authentication

4.1 Authentication Methods

FIPS 140-3 does not require an authentication mechanism for level 1 modules. Therefore, the module does not support an authentication mechanism for Crypto Officer. The Crypto Officer role is authorized to access all services provided by the module (see Table - Approved Services and Table - Non-Approved Services).

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	CO	None

Table 48: Roles

4.3 Approved Services

The module implements a dedicated API function to indicate if a requested service utilizes an approved security function (see also section 2.4).

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
AES Encryption/Decryption	Execute AES-mode encrypt or decrypt operation	1	plaintext data and key / ciphertext data and key	ciphertext data / plaintext data	Unauthenticated Symmetric Encryption and Decryption Authenticated Symmetric Encryption and Decryption	Crypto Officer - AES key: W,E
AES Key Wrapping / Key unwrapping	Execute AES-key wrapping or unwrapping operation	1	AES key wrapping key, unwrapped key / Wrapped key, AES key	wrapped key / unwrapped key	Key wrapping/ Key unwrapping	Crypto Officer - AES key-wrapping key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
			wrapping key			
Secure Hash Generation	Generate a digest for the requested algorithm	1	message	digest	Message Digest	Crypto Officer
Message Authentication Generation	Generate a MAC digest using the requested SHA algorithm or AES algorithm	1	message, MAC key, MAC algorithm	MAC	CMAC Message Authentication HMAC Message Authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
Message Authentication Code Verification	Verify a MAC digest	1	MAC, message, MAC key, MAC algorithm	pass/fail	CMAC Message Authentication HMAC Message Authentication	Crypto Officer - AES key: W,E - HMAC key: W,E
RSA signature generation and verification	Sign a message with a specified RSA private key. Verify the signature of a message with a specified RSA	1	SigGen: private key, message, hash function; SigVer: public key, digital signature, message,	SigGen: computed signature; SigVer: pass/fail result of digital signature verification	RSA Digital Signature Generation RSA Digital Signature Verification	Crypto Officer - RSA key pair: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	public key.		hash function			
ECDSA signature generation and verification	Sign a message with a specified ECDSA private key Verify the signature of a message with a specified ECDSA public key	1	SigGen: private key, message, hash function; SigVer: public key, digital signature, message, hash function	SigGen: computed signature; SigVer: pass/fail result of digital signature verification	ECDSA Digital Signature Generation ECDSA Digital Signature Verification	Crypto Officer - ECDSA key pair: W,E
Random Number Generation	Generate random number	1	Length or size for requested numbers	random bit-string	Random Number Generation	Crypto Officer - Entropy input string: E - DRBG seed, internal state V value, and key: G,R,E
PBKDF	Derive key from password	1	Password	PBKDF derived key	PBKDF Key Derivation	Crypto Officer - PBKDF derived key: G,R - PBKDF password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
KBKDF	Derive key from key derivation key	1	Derivation key	KBKDF derived key	KBKDF Key Derivation with HMAC KBKDF Key Derivation with CMAC	Crypto Officer - KBKDF key derivation key: W,E - KBKDF derived key: G,R,E
RSA key pair generation	Generate a keypair for a requested modulus	1	Modulus size	RSA key pair	RSA Asymmetric Key Generation	Crypto Officer - RSA key pair: G,R,E
ECDSA key pair generation	Generate a keypair for a requested elliptic curve	1	Curve	ECDSA key pair	ECDSA Asymmetric Key Generation ECDSA Asymmetric Key Verification	Crypto Officer - ECDSA key pair: G,R,E
Safe primes key generation	Generate a keypair for a requested 'safe' domain parameter	1	Curve	Diffie_Hellman key pair	Safeprime Key Generation	Crypto Officer - Diffie-Hellman key pair: G,R,E
Diffie-Hellman shared secret computation	Generate a shared secret	1	Received public key and possessed private key	DH shared secret	FFC Shared Secret Computation	Crypto Officer - Diffie-Hellman shared

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						secret: G,R,W,E
EC Diffie-Hellman shared secret computation	Generate a shared secret	1	Received public key and possessed private key	ECDH shared secret	ECC Shared Secret Computation	Crypto Officer - EC Diffie-Hellman shared secret: G,R,W,E
Self-test	execute CASTs in table section 10.2	1	power	pass/fail results	Unauthenticated Symmetric Encryption and Decryption Authenticated Symmetric Encryption and Decryption Random Number Generation ECDSA Asymmetric Key Generation ECDSA Asymmetric Key Verification ECDSA Digital Signature Generation ECDSA Digital Signature Verification CMAC	Crypto Officer - HMAC key: E - AES key: E - AES key-wrapping key: E - ECDSA key pair: E - RSA key pair: E - DRBG seed, internal state V value, and key: E - PBKDF derived key: E - KBKDF key derivati

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					Message Authentication HMAC Message Authentication ECC Shared Secret Computation FFC Shared Secret Computation KBKDF Key Derivation with HMAC Key wrapping/ Key unwrapping RSA Asymmetric Key Generation RSA Digital Signature Generation RSA Digital Signature Verification Safeprime Key Generation Message Digest PBKDF Key Derivation KBKDF Key Derivation with CMAC	on key: E - KBKDF derived key: E - EC Diffie-Hellman shared secret: E - EC Diffie Hellman key pair: E - Diffie-Hellman shared secret: E - Diffie-Hellman key pair: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show Status	Return the module status	N/A	N/A	Status output	None	Crypto Officer
Show module version info	Return Module Base Name and Module Version Number	N/A	N/A	Module informatioin	None	Crypto Officer
Zeroization	SSPs are zeroised when the system is powered down, when all resources of symmetric crypto function context, all resources of hash context, all resources of Diffie-Hellman context for Diffie-Hellman and EC Diffie-Hellman,	1	N/A	N/A	None	Crypto Officer - AES key: Z - AES key-wrapping key: Z - HMAC key: Z - ECDSA key pair: Z - RSA key pair: Z - Entropy input string: Z - DRBG seed, internal state V value, and key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	all resources of asymmetric crypto function context and all resources of key derivation function context are released					- PBKDF derived key: Z - KBKDF key derivation key: Z - PBKDF password: Z - KBKDF key derivation key: Z - Diffie-Hellman key pair: Z - EC Diffie Hellman key pair: Z - Diffie-Hellman shared secret: Z - EC Diffie-Hellman shared secret: Z

Table 49: Approved Services

The abbreviations of the access rights to SSPs have the following interpretation:

G = **Generate**: The module generates or derives the SSP.
R = **Read**: The SSP is read from the module (e.g., the SSP is output).
W = **Write**: The SSP is updated, imported, or written to the module.
E = **Execute**: The module uses the SSP in performing a cryptographic operation.
Z = **Zeroise**: The module zeroises the SSP.
N/A = The service does not access any SSP during its operation

4.4 Non-Approved Services

Name	Description	Algorithms	Role
Triple-DES encryption / decryption	Execute Triple-DES mode encrypt or decrypt operation.	Triple-DES [SP 800-67r2]	CO
RSA Key Encapsulation	The CAST does not perform the full KTS, only the raw RSA encrypt/decrypt.	RSA Key Wrapping	CO
RSA Key pair Generation	Generate a keypair with non-approved key sizes	RSA KeyGen	CO
RSA Signature Generation	Sign a message with non-approved private key	RSA SigGen	CO
RSA Signature Verification	Verify the signature of a message with a non-approved public key.	RSA SigVer	CO
Diffie Hellman Shared Secret Computation	For key sizes < 2048	Diffie-Hellman	CO
EC Diffie Hellman Shared Secret Computation	For curve sizes < P-224	EC Diffie-Hellman	CO
ECDSA key-pair generation , ECDSA key verification, ECDSA signature generation, ECDSA signature verification	For curve P-192	ECDSA	CO
ECDSA Key Pair Generation for compact point representation of points	For compact point representation of points	ECDSA KeyGen	CO
EdDSA Key Generation, Signature Generation, Signature Verification	Ed25519	EdDSA	CO
ECDH Key Agreement	X25519	ECDH	CO

Name	Description	Algorithms	Role
Hybrid encryption scheme	Encryption schemes that combine asymmetric and symmetric algorithms	Integrated Encryption Scheme on elliptic curves (ECIES) HPKE (Hybrid Public Key Encryption) [RFC9180]	CO
ANSI X9.63 Key Derivation	SHA-1 hash-based	ANSI X9.63 KDF	CO
SP800-56Crev2 Key Derivation (HKDF)	SHA-256 hash-based	HKDF [SP800-56Crev2]	CO
RFC6637 Key Derivation	SHA hash based	RFC6637	CO
OMAC Message Authentication Code Generation and Verification	One-Key CBC-MAC using 128-bit key	OMAC (One-Key CBC MAC)	CO
Message digest generation	Message digest generation using non-approved algorithms	MD2 MD4 MD5 (except in the TLS 1.0/1.1 context) RIPEMD SHA-3 Keccak	CO
Symmetric encryption / decryption	Symmetric encryption / decryption using non-approved algorithms	Blowfish CAST5 DES RC2 RC4	CO

Table 50: Non-Approved Services

4.5 External Software/Firmware Loaded

N/A

5 Software/Firmware Security

5.1 Integrity Techniques

A software integrity test is performed on the runtime image of the module. The HMAC-SHA256 implemented in the module is used as the approved algorithm for the integrity test. If the test fails, the module enters an error state where no cryptographic services are provided, and data output is prohibited i.e. the module is not operational.

5.2 Initiate on Demand

The module's integrity test can be performed on demand by power-cycling the computing platform. Integrity tests on demand is performed as part of the Pre-Operational Self-Tests. It is automatically executed at power- on.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Modifiable

6.2 Configuration Settings and Restrictions

The module is supplied as part of Device OS, a commercially available general-purpose operating system executing on the computing platforms specified in section 2.2.

7 Physical Security

The FIPS 140-3 physical security requirements do not apply to the Apple corecrypto Module v12.0 [Apple silicon, User, Software, SL1] since it is a software module.

8 Non-Invasive Security

Per IG 12.A, until the requirements of NIST SP 800-140F are defined, non-invasive mechanisms fall under ISO/IEC 19790:2012 Section 7.12 Mitigation of other attacks.

The requirements of this area are not applicable to the module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	The module stores ephemeral SSPs in RAM provided by the operational environment. They are received for use or generated by the module only at the command of the calling application. The operating system protects all SSPs through the memory separation and protection mechanisms. No process other than the module itself can access the SSPs in its process' memory.	Dynamic

Table 51: Storage Areas

9.2 SSP Input-Output Methods

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

Table 52: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Context object destruction	SSPs are zeroised when the appropriate context object is destroyed	Zeroization when structure is deallocated	By calling the zeroization function cc_clear
Power down	SSPs are zeroised when the system is powered down	SSPs are zeroised when the system is powered down	Operator can initiate power down

Zeroization Method	Description	Rationale	Operator Initiation
Intermediate value zeroization	Intermediate keygen values are zeroized before the module returns from the key generation function.	Intermediate keygen values are zeroized before the module returns from the key generation function.	N/A

Table 53: SSP Zeroization Methods

Data output interfaces are inhibited while zeroisation is performed.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES key	AES key	128 to 256 bits - 128 to 256 bits	Symmetric - CSP			Unauthenticated Symmetric Encryption and Decryption Authenticated Symmetric Encryption and Decryption CMAC Message Authentication
AES key-wrapping key	AES KW	128 to 256 bits - 128 to 256 bits	symmetric - CSP			Key wrapping/ Key unwrapping
HMAC key	HMAC key	8 - 262144 bits - 112 to 256-bits	MAC - CSP			HMAC Message Authentication
ECDSA key pair	ECDSA key pair (including intermediate keygen values)	P-224, P-256, P-384, P-521 - 112 to 256 bits	Asymmetric - CSP	ECDSA Asymmetric Key Generation		ECDSA Asymmetric Key Verification ECDSA Digital Signature Generation ECDSA Digital

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Signature Verification
RSA key pair	RSA key pair (including intermediate keygen values)	2048 - 4096 - 112 to 150 bits	Asymmetric - CSP	RSA Asymmetric Key Generation		RSA Digital Signature Generation RSA Digital Signature Verification
Entropy input string	Entropy input string. Obtained from the entropy source, used to seed the DRBG	256 bits - 256 bits	Entropy input string - CSP			Random Number Generation
DRBG seed, internal state V value, and key	DRBG input parameters	256 bits - 256 bits	DRBG - CSP	Random Number Generation		Random Number Generation
PBKDF derived key	PBKDF derived key	128 to 256 bits - 128 to 256 bits	Storage key - CSP	PBKDF Key Derivation		
PBKDF password	PBKDF password	64 to 1024 bits - N/A	Password - CSP			PBKDF Key Derivation
KBKDF key derivation key	KBKDF key derivation key	128 to 256 bits - 128 to 256 bits	Derivation key - CSP			KBKDF Key Derivation with HMAC KBKDF Key

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
						Derivation with CMAC
KBKDF derived key	KBKDF derived key	128 to 256 bits - 128 to 256 bits	Derived key - CSP	KBKDF Key Derivation with HMAC KBKDF Key Derivation with CMAC		
Diffie-Hellman key pair	Diffie-Hellman key pair (including intermediate keygen values)	MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112 to 200 bits	Asymmetric - CSP	Safeprime Key Generation		FFC Shared Secret Computation
Diffie-Hellman shared secret	Diffie-Hellman shared secret	MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 - 112 to 200 bits	Asymmetric - CSP		FFC Shared Secret Computation	
EC Diffie Hellman key pair	EC Diffie-Hellman key pair (including intermediat	P-224, P-256, P-384, P-521 -	Asymmetric - CSP	ECDSA Asymmetric Key Generation		ECC Shared Secret Computation

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	e keygen values)	112-256 bits				
EC Diffie-Hellman shared secret	EC Diffie-Hellman shared secret	P-224, P-256, P-384, P-521 - 112-256 bits	Asymmetric - CSP		ECC Shared Secret Computation	

Table 54: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES key	API input parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	
AES key-wrapping key	API input parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	
HMAC key	API input parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	
ECDSA key pair	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down Intermediate value zeroization	DRBG seed, internal state V value, and key:Used With
RSA key pair	API input parameters	RAM:Plaintext	From service invocation	Context object destruction Power down	DRBG seed, internal state V

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
	API output parameters		to service completion	Intermediate value zeroization	value, and key:Used With
Entropy input string		RAM:Plaintext	Storage duration during the usage of the CSP	Power down	DRBG seed, internal state V value, and key:Used With
DRBG seed, internal state V value, and key		RAM:Plaintext	Storage duration during the usage of the CSP	Power down	Entropy input string:Derived From
PBKDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	PBKDF password:Derived From
PBKDF password	API input parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	PBKDF derived key:Used With
KBKDF key derivation key	API input parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	KBKDF derived key:Used With
KBKDF derived key	API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	KBKDF key derivation key:Derived From
Diffie-Hellman key pair	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down Intermediate value zeroization	Diffie-Hellman shared secret:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Diffie-Hellman shared secret	API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	Diffie- Hellman key pair:Used With
EC Diffie Hellman key pair	API input parameters API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down Intermediate value zeroization	EC Diffie-Hellman shared secret:Used With
EC Diffie-Hellman shared secret	API output parameters	RAM:Plaintext	From service invocation to service completion	Context object destruction Power down	EC Diffie Hellman key pair:Used With

Table 55: SSP Table 2

10 Self-Tests

While the module is executing the self-tests, services are not available, and input and output are inhibited.

10.1 Pre-Operational Self-Tests

The module performs a pre-operational software integrity automatically when the module is loaded into memory (i.e., at power on) before the module transitions to the operational state. A software integrity test is performed on the runtime image of the module with HMAC-SHA256 used to perform the approved integrity technique. Prior to using HMAC-SHA-256, a Conditional Cryptographic Algorithm Self-Tests (CAST) is performed.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A3486)	112-bit key	Message Authentication	SW/FW Integrity	Module successful execution	The HMAC-SHA2-256 value calculated at runtime is compared with the HMAC-SHA2-256 value stored in the module, computed at compilation time.

Table 56: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A3424)	128-bit key	KAT	CAST	Module becomes operational	Authenticated decryption operation	Test runs at Power-on before the integrity test
Counter DRBG (A3487)	AES 128-bit key	KAT	CAST	Module becomes operational	Health test per SP800- 90ARev1 section 11.3	Test runs at Power-on before the integrity test
HMAC-SHA2-256 (A3426)	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA-1 (A3426)	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
HMAC-SHA2-512 (A3426)	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
RSA KeyGen (FIPS186-4) (A3426)	SHA2-256 and respective keys	PCT	PCT	Successful key generation	Calculation and verification of a digital signature	RSA key pair generation.
RSA SigGen (FIPS186-4) (A3426)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Signature Generation or Key Generation service request	Test runs at Power-on before the integrity test
RSA SigVer (FIPS186-4) (A3426)	PKCS#1 v1.5 with 2048 bit key and SHA2-256	KAT	CAST	Module becomes operational	Signature Verification or Key Generation service request	Test runs at Power-on before the integrity test
ECDSA KeyGen (FIPS186-4) (A3426)	SHA2-256 and respective keys	PCT	PCT	Successful key generation	Key generation	EC key pair generation.
ECDSA SigGen (FIPS186-4) (A3426)	P-224 with SHA-224	KAT	CAST	Module becomes operational	Signature Generation or Key Generation service request	Test runs at Power-on before the integrity test
ECDSA SigVer (FIPS186-4) (A3426)	P-224 with SHA-224	KAT	CAST	Module becomes operational	Signature Verification or Key Generation service request	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 (A3426)	P-224 curve	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at Power-on before the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A3426)	MODP-2048	KAT	CAST	Module becomes operational	Shared secret computation	Test runs at Power-on before the integrity test
KDF SP800-108 (A3426)	Counter mode using SHA-1, SHA-256, SHA-512	KAT	CAST	Module becomes operational	Key derivation	Test runs at Power-on before the integrity test
PBKDF (A3426)	SHA-1, SHA-256, SHA-512	KAT	CAST	Module becomes operational	Key derivation	Test runs at Power-on before the integrity test
Safe Primes Key Generation (A3426)	MODP-2048	PCT	PCT	Successful key generation	Section 5.6.2.1.4 of SP 800-56Arev3	key gen
AES-CBC (A3423)	128-bit key	KAT	CAST	Module becomes operational	Encryption and decryption run separately	Test runs at Power-on before the integrity test
AES-ECB (A3423)	128-bit key	KAT	CAST	Module becomes operational	Encryption and decryption run separately	Test runs at Power-on before the integrity test
AES-XTS Testing Revision 2.0 (A3483)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CCM (A3424)	128-bit key	KAT	CAST	Module becomes operational	Authenticated encryption and	Test runs at Power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
					decryption run separately	before the integrity test
AES-CMAC (A3426)	128-bit key	KAT	CAST	Module becomes operational	Authenticated encryption	Test runs at Power-on before the integrity test
HMAC-SHA2-512/256 (A3486)	SHA2-512/256	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test

Table 57: Conditional Self-Tests

10.3 Periodic Self-Test Information

None

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error State	1) The HMAC-SHA-256 value computed over the module did not match the pre-computed value or 2) The computed value in the invoked Conditional CAST did not match the known value or 3) The	1) Pre-operational Software Integrity Test failure or 2) Conditional CAST failure 3) Conditional PCT failure	Power cycle the device which results in the module being reloaded into memory and reperforming the pre-operational software integrity test and the Conditional CASTs.	1) Print statement "FAILED: fipspost_post_integrity" to stdout or 2) Print statement "FAILED:<event>" to stdout (<event> refers to any of the cryptographic functions listed Table - Conditional Self-Tests 3) Error code "CCEC_GENERATE_KEY_CONSISTENCY" returned for ECDSA and EC Diffie-Hellman Error code "CCRSA_GENERATE_KEY_CONSISTENCY" returned for RSA Error code "CCDH_GENERATE_KEY_CONSISTENCY" returned for Diffie-Hellman

Name	Description	Conditions	Recovery Method	Indicator
	signature failed to verify successfully in the Conditional PCT. No cryptographic services are provided, and data output is prohibited			

Table 58: Error States

10.5 Operator Initiation of Self-Tests

The module permits operators to initiate the pre-operational or conditional self-tests on demand for periodic testing of the module by rebooting the system (i.e., power-cycling).

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Startup Procedures: The module is built into Device OS defined in section 2 and delivered/ installed with the respective Device OS. There is no standalone delivery of the module as a software library.

Installation Process and Authentication Mechanisms: The vendor's internal development process guarantees that the correct version of module goes with its intended Device OS version. For additional assurance, the module is digitally signed by vendor, and it is verified during the integration into Host Device OS.

This digital signature-based integrity protection during the delivery/integration process is not to be confused with the HMAC-256 based integrity check performed by the module itself as part of its pre-operational self- tests.

11.2 Administrator Guidance

The Approved mode of operation is configured in the system by default and can only be transitioned into the non-Approved mode by calling one of the non-Approved services listed in Table - Non-Approved Services. If the device starts up successfully, then the module has passed all self-tests and is operating in the Approved mode.

The ESV Public Use Document (PUD) reference for physical entropy source is: https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E14_PublicUse.pdf

The ESV Public Use Document (PUD) reference for non-physical entropy source is: https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/entropy/E15_PublicUse.pdf

Apple Platform Certifications guide [platform certifications] and Apple Platform Security guide [SEC] are provided by Apple which offers IT System Administrators with the necessary technical information to ensure FIPS 140-3 Compliance of the deployed systems. This guide walks the reader through the system's assertion of cryptographic module integrity and the steps necessary if module integrity requires remediation.

11.3 Non-Administrator Guidance

None

11.4 Design and Rules

The Crypto Officer shall consider the following requirements and restrictions when using the module.

- AES-GCM see section 2.7.
- AES-XTS see section 2.7.
- PBKDF see section 2.7.

11.6 End of Life

The module secure sanitization is accomplished through the Lost Mode, remote wipe, and remote lock sections of the provided vendor document [platform certifications]. The operator can initiate sanitization.

12 Mitigation of Other Attacks

The module does not claim mitigation of other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
API	Application Programming Interfaces
CAST	Cryptographic Algorithm Self-Test
CAST5	A symmetric-key 64-bit block cipher with 128-bit key
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter Mode
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ESVP	Entropy Source Validation Program
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
KAS	Key Agreement Schema
KAT	Known Answer Test
KBKDF	Key Based Key Derivation Function
KDF	Key Derivation Function
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
OAEP	Optimal Asymmetric Encryption Padding
OFB	Output Feedback
PAA	Processor Algorithm Acceleration
PBKDF	Password Based Key Derivation Function
PRF	Pseudo-Random Function
PSS	Probabilistic Signature Scheme
PUD	Public Use Document (ESVP)
RSA	Rivest, Shamir, Addleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSC	Shared Secret Computation
TOEPP	Tested Operational Environment Physical Perimeter
XTS	XEX-based Tweaked-codebook mode with cipher text Stealing

Appendix B. References

FIPS140-3	**FIPS PUB 140-3 - Security Requirements for Cryptographic Modules** March 2019 https://doi.org/10.6028/NIST.FIPS.140-3
SP 800-140x	**CMVP FIPS 140-3 Related Reference** https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-standards
FIPS140-3_IG	**Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program** September 2020 https://csrc.nist.gov/Projects/cryptographic-module-validation-program/fips-140-3-ig-announcements
FIPS140-3_MM	**CMVP FIPS 140-3 Draft Management Manual** https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS-140-3-CMVP%20Management%20Manual%20v2.0.pdf
SP 800-140	**FIPS 140-3 Derived Test Requirements (DTR)** https://csrc.nist.gov/publications/detail/sp/800-140/final
SP 800-140A	**CMVP Documentation Requirements** https://csrc.nist.gov/publications/detail/sp/800-140a/final
SP 800-140Br1	**CMVP Security Policy Requirements** https://csrc.nist.gov/pubs/sp/800/140/b/r1/final
SP 800-140C	**CMVP Approved Security Functions** https://csrc.nist.gov/publications/detail/sp/800-140c/final
SP 800-140D	**CMVP Approved Sensitive Security Parameter Generation and Establishment Methods** https://csrc.nist.gov/publications/detail/sp/800-140d/final
SP 800-140E	**CMVP Approved Authentication Mechanisms** https://csrc.nist.gov/publications/detail/sp/800-140e/final
SP 800-140F	**CMVP Approved Non-Invasive Attack Mitigation Test Metrics** https://csrc.nist.gov/publications/detail/sp/800-140f/final
FIPS180-4	**Secure Hash Standard (SHS)** March 2012 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
FIPS186-4	**Digital Signature Standard (DSS)** July 2013 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
FIPS197	**Advanced Encryption Standard** November 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

FIPS198-1	**The Keyed Hash Message Authentication Code (HMAC)** July 2008 http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf
FIPS202	**SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions** August 2015 http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
PKCS#1	**Public Key Cryptography Standards (PKCS) #1: RSA Cryptography** Specifications Version 2.1 February 2003 http://www.ietf.org/rfc/rfc3447.txt
RFC3394	**Advanced Encryption Standard (AES) Key Wrap Algorithm** September 2002 http://www.ietf.org/rfc/rfc3394.txt
RFC5649	**Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm** September 2009 http://www.ietf.org/rfc/rfc5649.txt
RFC9180	**Hybrid Public Key Encryption** February 2022 https://www.ietf.org/rfc/rfc9180.pdf
SP800-38A	**NIST Special Publication 800-38A - Recommendation for Block Cipher Modes of Operation Methods and Techniques** December 2001 http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf
SP800-38B	**NIST Special Publication 800-38B - Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication** May 2005 http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
SP800-38C	**NIST Special Publication 800-38C - Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality** May 2004 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
SP800-38D	**NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC** November 2007 http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
SP800-38E	**NIST Special Publication 800-38E - Recommendation for Block Cipher Modes of Operation: The XTS AES Mode for Confidentiality on Storage Devices** January 2010 http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf
SP800-38F	**NIST Special Publication 800-38F - Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping** December 2012 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf

SP800-38G	**NIST Special Publication 800-38G - Recommendation for Block Cipher Modes of Operation: Methods for Format - Preserving Encryption** March 2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf
SP800-56Ar3	**Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography** April, 2018 https://doi.org/10.6028/NIST.SP.800-56Ar3
SP800-56Br2	**Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography** March 2019 https://doi.org/10.6028/NIST.SP.800-56Br2
SP800-56Cr2	**Recommendation for Key-Derivation Methods in Key-Establishment Schemes** August 2020 https://doi.org/10.6028/NIST.SP.800-56Cr2
SP800-57	**NIST Special Publication 800-57 Part 1 Revision 5 - Recommendation for Key Management Part 1: General** May 2020 https://doi.org/10.6028/NIST.SP.800-57pt1r5
SP800-67r2	**NIST Special Publication 800-67 Revision 1 - Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher** January 2012 (withdrawn January 2014) https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf
SP800-90Ar1	**NIST Special Publication 800-90A - Revision 1 - Recommendation for Random Number Generation Using Deterministic Random Bit Generators** June 2015 http://dx.doi.org/10.6028/NIST.SP.800-90Ar1
SP800-90B	**NIST Special Publication 800-90B - Recommendation for the Entropy Sources Used for Random Bit Generation** January 2018 https://doi.org/10.6028/NIST.SP.800-90B
SP800-108r1	**NIST Special Publication 800-108 - Recommendation for Key Derivation Using Pseudorandom Functions (Revision 1)** https://doi.org/10.6028/NIST.SP.800-108r1
SP800-131Ar2	**Transitioning the Use of Cryptographic Algorithms and Key Lengths** March 2019 https://doi.org/10.6028/NIST.SP.800-131Ar2
SP800-132	**NIST Special Publication 800-132 - Recommendation for Password-Based Key Derivation - Part 1: Storage Applications** December 2010 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf

SP800-133r2	**Recommendation for Cryptographic Key Generation** June 2020 https://doi.org/10.6028/NIST.SP.800-133r2
SP800-135r1	**NIST Special Publication 800-135 Revision 1 - Recommendation for Existing Application-Specific Key Derivation Functions** December 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-135r1.pdf
SEC	**Apple Platform Security** https://support.apple.com/guide/security/welcome/web https://manuals.info.apple.com/MANUALS/1000/MA1902/en_US/apple-platform-security-guide.pdf
platform certifications	**Apple Security Certifications and Compliance Center** https://support.apple.com/en-gw/guide/certifications/welcome/web