Apple Inc.

Apple corecrypto Module v12.0 [Apple silicon, Secure Key Store, Hardware, SL2/PHY3]

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 v12.0 [Apple silicon, Secure Key Store, Hardware, SL2/PHY3] cryptographic module. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 2 module.

This document provides all tables and diagrams (when applicable) required by NIST SP 800-140Br1.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use: The Apple corecrypto Module v12.0 [Apple silicon, Secure Key Store, Hardware, SL2/PHY3] cryptographic module (hereafter referred to as "the module") consists of both firmware and hardware components. The Secure Key Store (SKS) application is the module's firmware which operates within the sepOS execution environment which is separate from the Device OS' (iPadOS 15) execution environment. The firmware interface is defined as the API offered by the module's mailbox interface to callers from the Device OS execution environment. SKS has an API layer that provides consistent interfaces to the supported services and therefore the supported cryptographic algorithms. In addition, the module provides Inter-Process Communication (IPC) interfaces to other applications executing within the sepOS execution environment. The sepOS execution environment is driven by its own CPU and operates from a dedicated region of the device's memory. Both the Device's and sepOS' execution environments are physically separated on the SoC and thus execute independently of each other.

Module Type: Hardware

Module Embodiment: SingleChip

Module Characteristics: SubChip

Cryptographic Boundary: The module cryptographic boundary is delineated by the dotted blue rectangle in the Figure 1. The cryptographic module boundary includes the following hardware components:

- Hardware Random Number Generator composed of a SP800-90A Approved CTR_DRBG and a physical entropy source compliant to SP800-90B.
- Hardware AES implementing AES-ECB and AES-CBC encryption and decryption.
- Hardware Public Key Accelerator (PKA) used for generating asymmetric key pairs.
- A volatile RAM for storing runtime SSPs.
- A non-volatile Flash for storing an encrypted Class D key.

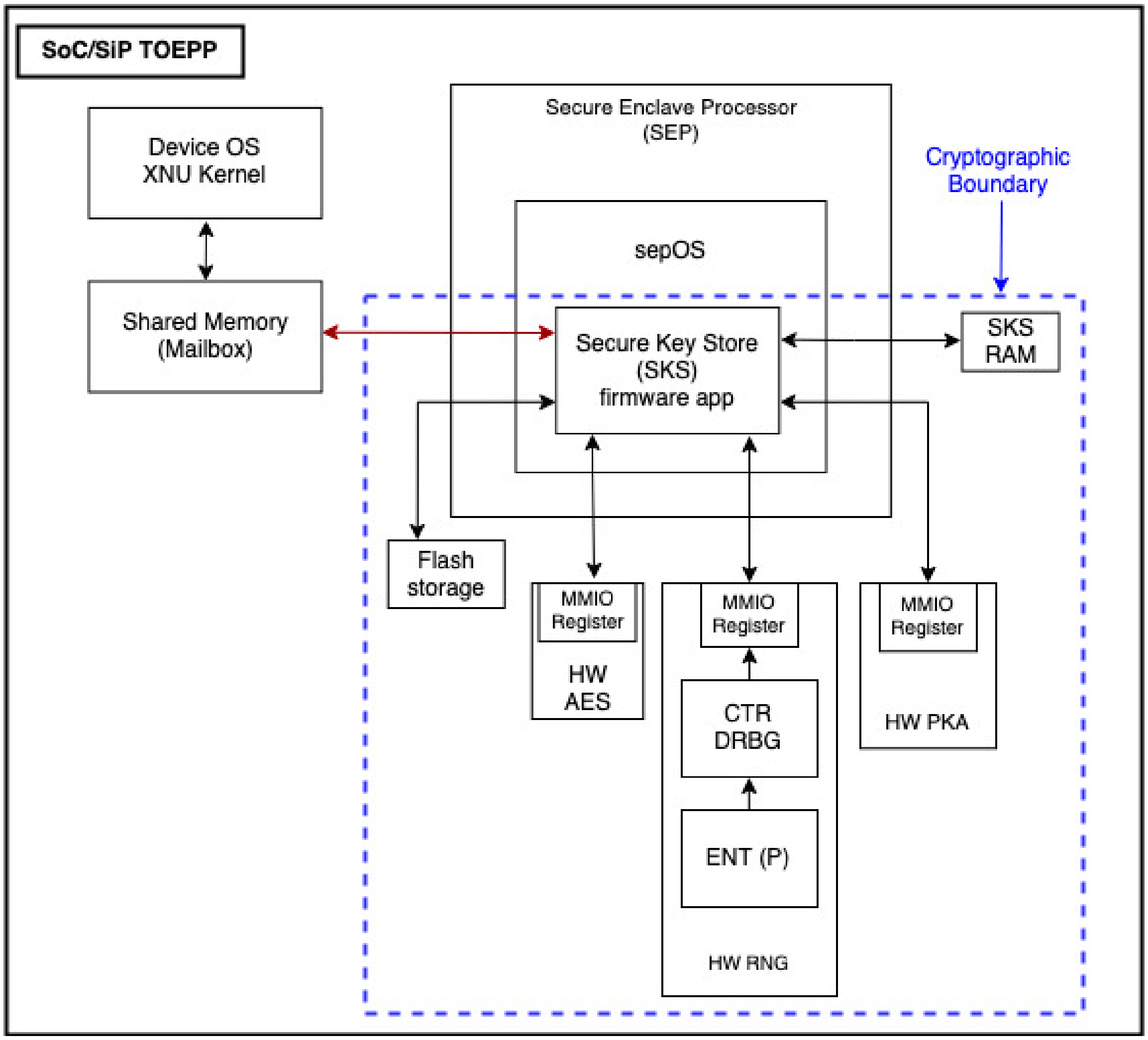

Figure 1: Block Diagram

Tested Operational Environment's Physical Perimeter (TOEPP): The physical perimeter is represented by the most exterior black line in the block diagram Figure 1. A photograph of each hardware module is shown below

Figure 2: Apple A Series A13 Bionic

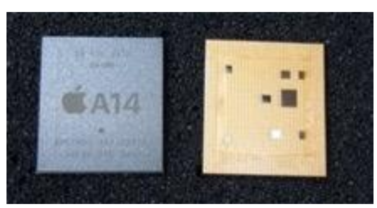

Figure 3: Apple A Series A14 Bionic

Figure 4: Apple A Series A15 Bionic

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
SKS on A13 Bionic embedded in iPad (9th generation) running sepOS distributed with iPadOS 15	2.0	12.0	Apple A Series A13 Bionic	N/A
SKS on A14 Bionic embedded in iPad (4th generation) running sepOS distributed with iPadOS 15	2.0	12.0	Apple A Series A14 Bionic	N/A
SKS on A15 Bionic embedded in iPad (6th generation) running sepOS distributed with iPadOS 15	2.0	12.0	Apple A Series A15 Bionic	N/A

Table 2: Tested Module Identification – Hardware

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	return 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	return 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 3: Modes List and Description

2.5 Algorithms

Approved Algorithms:

AES-CBC

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A1469	Direction - Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2842	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2843	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2844	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2845	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2863	Direction - Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A510	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Table 4: Approved Algorithms - AES-CBC

AES-ECB

Algorithm	CAVP Cert	Properties	Reference
AES-ECB	A1362	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	A1469	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2842	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2843	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2845	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2847	-	SP 800-38A
AES-ECB	A2863	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2864	Direction - Encrypt Key Length - 256	SP 800-38A

Algorithm	CAVP Cert	Properties	Reference
AES-ECB	A501	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	A510	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A

Table 5: Approved Algorithms - AES-ECB

AES-KW

Algorithm	CAVP Cert	Properties	Reference
AES-KW	A2843	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A2845	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A2846	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F

Table 6: Approved Algorithms - AES-KW

CTR_DRBG

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A1362	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	A2864	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	A501	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1

Table 7: Approved Algorithms - CTR_DRBG

HMAC

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA-1	A2845	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA-1	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-224	A2845	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-224	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A2845	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A2849	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-384	A2845	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-384	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512	A2845	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A2848	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1

Table 8: Approved Algorithms - HMAC

Message Digest

Algorithm	CAVP Cert	Properties	Reference
SHA-1	A2845	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA-1	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-224	A2845	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-224	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-256	A2845	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-256	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-256	A2849	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4

Algorithm	CAVP Cert	Properties	Reference
SHA2-384	A2845	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-384	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512	A2845	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512/256	A2848	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4

Table 9: Approved Algorithms - Message Digest

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric	N/A	SP800-133 Rev2 Section 4, example 1

Table 10: Vendor-Affirmed Algorithms

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
Ed25519 Key generation	EdDSA signature scheme
Ed25519 shared secret generation	EdDSA shared secret generation
Curve 25519 key generation	key generation
Curve 25519 shared secret generation	shared secret generation
ECDH Key Pair Generation	Elliptic Curve Integrated Encryption Scheme (ECIES) Key Generation
ECDH Shared Secret Computation	Elliptic Curve Integrated Encryption Scheme (ECIES) Encryption/Decryption
ANSI X9.63 KDF	Elliptic Curve Integrated Encryption Scheme (ECIES) Encryption/Decryption
AES-GCM	Elliptic Curve Integrated Encryption Scheme (ECIES) Encryption/Decryption
HKDF RFC5869	HMAC based Key Derivation Function
PBKDF	Key Derivation

Name	Use and Function
ECDSA implemented in FW	Key generation as part of Ref key generation service and validation, Signature generation and verification as part of Device keybag service
ECDSA implemented in HW PKA	Key generation as part of Ref key generation service Signature generation primitive
ECDH implemented in FW	Shared secret computation
ECDH implemented in HW PKA	Shared secret computation
AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Key wrapping and unwrapping

Table 11: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Unauthenticated Symmetric Encryption and Decryption	BC-UnAuth	AES Encrypt/Decrypt	AES [FIPS 197; SP 800-38A]:CBC, ECB	AES-CBC: (A2842, A2843, A2844, A2845, A510, A1469, A2863) Key Size/Strength: 128, 192, 256 AES-ECB: (A2842, A2843, A2845, A510, A2847, A1469, A2863, A2864, A1362) Key Size/Strength: 128, 192, 256 AES-ECB: (A501) Key Size/Strength: 256
key wrapping / key unwrapping	KTS-Wrap	AES Key Wrapping	KTS (AES) [SP 800-38F]:AES-KW	AES-KW: (A2843, A2845, A2846) Key

Name	Type	Description	Properties	Algorithms
				Size/Strength: 128, 192, 256
Random Number Generation	DRBG	Random number generator using AES-256	CTR_DRBG [SP800-90ARev1]:AES-256; No Derivation Function; Prediction Resistance Enabled	Counter DRBG: (A501, A2864, A1362) Key Size/Strength: 256
HMAC Message Authentication	MAC	Key Length 8 - 262144 bits/ Key Strength: 112 to 256 bits	HMAC [FIPS 198]:SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256	HMAC-SHA-1: (A2845, A2848) HMAC-SHA2-224: (A2845, A2848) HMAC-SHA2-256: (A2845, A2848, A2849) HMAC-SHA2-384: (A2845, A2848) HMAC-SHA2-512: (A2845, A2848) HMAC-SHA2-512/256: (A2848)
Message Digest	SHA	Hash function	SHS [FIPS 180-4]:SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256	SHA-1: (A2845, A2848) SHA2-224: (A2845, A2848) SHA2-256: (A2845, A2848, A2849) SHA2-384: (A2845, A2848) SHA2-512: (A2845, A2848)

Name	Type	Description	Properties	Algorithms
				SHA2-512/256: (A2848)
Symmetric Key Generation	CKG	AES Key Generation	CKG [SP800-133Rev2]:Key Length/Key Strength: 256-bits	CKG: () AES key: Key Length/ Key Strength: 256 Counter DRBG: (A1362, A2864, A501)

Table 12: Security Function Implementations

2.7 Algorithm Specific Information

SHA-1:

The SHA-1 algorithm as implemented by the module will be non-approved for all purposes except signature verification, starting January 1, 2030.

2.8 RBG and Entropy

Cert Number	Vendor Name
E113	Apple

Table 13: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Apple corecrypto physical entropy source	Physical	See Tested Operational Environment Table	256 bit	256 bit	SHA-256 [ACVP cert. # C1223]

Table 14: Entropy Sources

Entropy sources : The internal physical noise source consisting of ring oscillators.

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

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

The deterministic random bit generators are seeded by the physical noise source.

RBG Output: The output of hardware entropy source provides 256-bits of security strength in instantiating and reseeding the module approved DRBGs.

2.9 Key Generation

See vendor affirmed algorithms (CKG) in section 2.5.

2.10 Key Establishment

The Module implements AES key wrapping and unwrapping as part of KTS in accordance with IG D.G method 2 and SP800-38F.

2.11 Industry Protocols

None for this module

3 Cryptographic Module Interfaces

The cryptographic interfaces of the module are provided through the mailbox interface that is used between the module and the Device OS kernel, and the IPC channel used between the module and other sepOS applications.

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Mailbox Memory, IPC channel	Data Input Data Output	Data inputs/outputs are provided through the memory used for mailbox and IPC
Mailbox Memory, IPC channel	Control Input	Control input which controls the module's operation is provided through the mailbox by the Device OS' kernel and to applications located within the sepOS execution environment through IPC.
Mailbox Memory, IPC channel	Status Output	Status output is provided in return codes and through messages returned via the mailbox or the IPC. Documentation for each service invocation 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 15: Ports and Interfaces

The module’s logical interfaces used for input data and control information are logically disconnected from the logical paths used for the output of data and status information by virtue of the module's API. The module’s API distinguishes all output data from SSP information.

The module does not implement a Control Output Logical Interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
AES-KW	Unwrapping function	key wrapping / key unwrapping	256-bits	60,000,000 * 1 / 2^256
Implicit	Implicit role assumption for non-crypto services	None	N/A	N/A

Table 16: Authentication Methods

Within the constraints of FIPS 140-3 overall security level 2 (with physical security at security level 3), the module implements a role-based authentication mechanism for authentication of the user role.

The module implements authenticated encryption-based mechanism in the following way: to request an authenticated service from the module the user must provide the credential and a reference to the class C or A keys of the user keybag that is stored encrypted under SP800-38F AES Key Wrapping (AES-KW) within the module. The module performs obfuscation on the Operator provided credential. The resulting value -called REK (Root Encryption Key)- is used as the 256-bit AES key. Using this key, the module decrypts all the class C or A keys in the referenced user keybag with SP800-38F AES Key Unwrapping function (i.e., AES-KW-AD). As AES-KW is an authentication cipher, the decryption operation will only succeed if there is no authentication error. If the user keybag can be successfully decrypted, the user is authenticated to the module and the requested crypto service will then be proceeded with the decrypted user key. The failure of decrypting the user keybag is also a user authentication failure and the Operator will be denied access to the module.

The User keybags are configured in the module during factory install. Each User keybag consists of set of class C, A and D keys. Specifically, class C keys include C key, CK key, CKU keys and the class A keys include A key, AK key, AKU key and APKU key. Only the class A or C keys are considered as approved. Any use of class D keys is considered as non-approved. The module maintains authenticated session from the time the User keybags are unwrapped until the power off. Upon power off, the unwrapped User keybags are zeroized and at the next power on the User credential needs to be provided again in order to unwrap the User keybag. All authentication data is provided electronically from the calling application/service and hence is not in visible form.

The module does not support concurrent operators.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
User	Role	Authenticated	AES-KW
Crypto Officer	Role	Non-authenticated	Implicit

Table 17: Roles

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
User Keybag Services via Mailbox	Step 1: The module receives User credential and the reference to the class C or A key from the User keybag; Step 2. Obfuscation is performed on the User provided credential resulting into a value called REK.; Step 3. REK is used as a key for the AES KW operation to unwrap the referenced class A or C keys in the user keybag stored in the module; Step 4. Status of unwrapping operation of class keys is returned via	Success returned from API listed in the customer proprietary guidance document	User credential, reference to class C/A key from the user keybag	status (success/error)	Unauthenticated Symmetric Encryption and Decryption key wrapping / key unwrapping	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - REK: W,E - Authentication Credential: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	mailbox interface and the REK is zeroized.					
General Authentication service	The module invokes the User Keybag Services via Mailbox (i.e. #1 above)	Success returned from API listed in the customer proprietary guidance document	User credential, reference to class C/A key from the user keybag	status (success/error)	key wrapping / key unwrapping	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - REK: W,E - Authentication Credential: W,E
Generation of Data Encryption Key (DEK)	Step 1: The module receives the reference to the class C or A key from the user keybag; Step 2: The module generates a new DEK using the DRBG; Step 3: Referenced class C or A key is used to wrap the DEK using AES-KW; Step 4: Wrapped DEK is	Success returned from API listed in the customer proprietary guidance document	reference to class C/A key from the User keybag	wrapped DEK	Symmetric Key Generation	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - Entropy input string: W,E - DRBG

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	sent out of the module					seed: W,E - DRBG internal state (V value, Key): W,E - Data Encryption Key (DEK) (AES key): G,W,E
Keychain DEK service using AK/AKU/AKPU/CK/CKU class key	Step 1. The module receives wrapped DEK (that was sent as part of service 3 above) and the pointer to class key AK/AKU/AKPU/CK/CKU from the user keybag; Step 2. Using the referenced class key, the module unwraps the DEK using AES-KW. If the class key is not available, an error is returned; Step 3. plaintext DEK is sent out to the User. (AS09.16)	Success returned from API listed in the customer proprietary guidance document	pointer to AK/AKU/AKPU/CK/CKU class key, wrapped DEK	unwrapped DEK	key wrapping / key unwrapping	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - Data Encryption Key (DEK) (AES key): W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Backup keybag generation	The module generates new set of back up keybags using the DRBG	Success returned from API listed in the customer proprietary guidance document	N/A	status (success/error)	Random Number Generation	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys): G,E - Entropy input string: W,E - DRBG seed: W,E - DRBG internal state (V value, Key): W,E
Backup keybag service	Step 1. The module receives wrapped DEK and the class key reference for C and A from the user keybag; 2. Using the referenced class key, the module unwraps the DEK using AES-KW. If the class key is not available, an	Success returned from API listed in the customer proprietary guidance document	wrapped DEK, reference to class C or A key from the user keybag	wrapped DEK	key wrapping / key unwrapping	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - Data Encryption Key (DEK)

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	error is returned; 3. The module generates a set of back up keybag using DRBG; 4. Unwrapped DEK is re-wrapped with back up keybag using AES-KW; 5. Wrapped DEK is sent out.					(AES key): W,E - Entropy input string: W,E - DRBG seed: W,E - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys): R,W,E - DRBG internal state (V value, Key): W,E
Escrow keybag creation	The module generates new set of escrow keybag using the DRBG	Success returned from API listed in the customer proprietary guidance document	N/A	status (success/error)	Random Number Generation	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Escrow Keybag (AES keys): G,E - Entropy input string:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W,E - DRBG seed: W,E - DRBG internal state (V value, Key): W,E
Export Keybag	Step 1. The module receives reference to a keybag; Step 2: A HMAC key is taken as input based on the hardware specific data for the SKS; Step 3: HMAC value is calculated on the entire referenced keybag that includes encrypted keys; Step 4: HMAC is appended at the end of the keybag; Step 5: keybag with the appended HMAC is output to the User	Success returned from API listed in the customer proprietary guidance document	reference to a keybag to be exported	keybag with HMAC tag	HMAC Message Authentication Message Digest	User - HMAC key: W,E - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): R,E - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys): R,E - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						in Escrow Keybag (AES keys): R,E
Device Wipe	Erase all content (Factory Reset)	Success returned from API listed in the customer proprietary guidance document	N/A	N/A	None	Crypto Officer - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): Z - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys): Z - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Escrow Keybag (AES keys): Z - Data

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Encryption Key (DEK) (AES key): Z - Entropy input string: Z - DRBG seed: Z - HMAC key: Z - Authentication Credential: Z - REK: Z - DRBG internal state (V value, Key): Z
Perform self test	Initiate pre-operational self-test and CASTs by powering off/on	N/A	module power-off/on	results of self-test	Unauthenticated Symmetric Encryption and Decryption key wrapping / key unwrapping Random Number Generation HMAC Message Authenticat	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
					ion Message Digest	
Show Status	N/A	N/A	N/A	status	None	Crypto Officer
Show Module Version Information	N/A	N/A	N/A	Module name and version	None	Crypto Officer

Table 18: 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
Class D key File System Services to wrap or unwrap DEK	Wrapping of provided plaintext DEK or unwrapping of provided wrapped DEK using class D key from Backup keybag or Flash in SEP	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Class D key service to encrypt or decrypt data	Encryption of provided plaintext or decryption of provided ciphertext using class D key from Device or iCloud Keybag	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer

Name	Description	Algorithms	Role
Class DK/DKU File System Services to wrap or unwrap keychain	Wrapping of provided plaintext keychain or unwrapping of provided wrapped keychain using class DK/DKU key from Backup keybag or User keybag	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Class DK/DKU key service for data encrypt or decrypt	Encryption of provided plaintext or decryption of provided ciphertext using DK/DKU key from Device or iCloud keybag	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Generate Ref-Key	Key Generation	Ed25519 Key generation Curve 25519 key generation ECDH Key Pair Generation	Crypto Officer
Sign and verify using Ref-key	Signature Generation and Verification	ECDSA implemented in FW ECDSA implemented in HW PKA	Crypto Officer
Encryption and decryption using Ref-key	shared secret is generated using user provided key and existing ref key followed by HKDF is applied to derived a key which is used to encrypt the provided plaintext or decrypt the provided ciphertext	AES-GCM HKDF RFC5869 ECDSA implemented in FW ECDSA implemented in HW PKA AES KW using class D key, keys from Device keybag, keys	Crypto Officer

Name	Description	Algorithms	Role
		from iCloud keybag or NVM storage controller key	
Generate Shared Secret using Ref-key	Shared secret generation	Ed25519 shared secret generation Curve 25519 shared secret generation ECDH Shared Secret Computation ECDH implemented in FW ECDH implemented in HW PKA	Crypto Officer
Device Keybag service for data encrypt or decrypt	Encryption of provided plaintext or decryption of provided ciphertext using any key from Device Keybag	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
iCloud Keybag service for data encrypt or decrypt	Encryption of provided plaintext or decryption of provided ciphertext using any key from iCloud Keybag	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Escrow keybag service for key wrapping and unwrapping	Wrapping of provided plaintext key or unwrapping of provided wrapped key using any key from Escrow Keybag	AES KW using class D key, keys from Device keybag, keys from iCloud	Crypto Officer

Name	Description	Algorithms	Role
		keybag or NVM storage controller key	
Encrypt or Decrypt service using Class B Curve 22519 key from any keybag	shared secret is computed by generating new ephemeral keypair and existing Curve25519 key followed by HKDF is applied to derived a key which is used doe data encryption or decryption. During encryption operations, the wrapped key and the ephemeral public key is sent to the user	Curve 25519 key generation Curve 25519 shared secret generation HKDF RFC5869 AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Wrap or unwrap service for DEK or keychain using any Curve 22519 key from asymmetric keybag	shared secret is computed by generating new ephemeral keypair and existing Curve25519 key followed by HKDF is applied to derived a key which is used to wrap and unwrap DEK or keychain. During wrapping operation, the wrapped key and the ephemeral public key is sent to the user	Curve 25519 key generation Curve 25519 shared secret generation HKDF RFC5869 AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Asymmetric (Ed25519) backup keybag wrap and unwrap	Pointer to DK/DKU/CK/CKU/AK/AKU/AKPU key from asymmetric keybag, plaintext keychain during wrapping operation or wrapped keychain during unwrapping operation	Ed25519 Key generation Ed25519 shared secret generation HKDF RFC5869 AES KW using class D key, keys from Device	Crypto Officer

Name	Description	Algorithms	Role
		keybag, keys from iCloud keybag or NVM storage controller key	
Wrap or unwrap service for keychain using DK/DKU/CK/ CKU/AK/AKU/AKPU Ed25519 key from asymmetric keybag	shared secret is computed by generating new ephemeral keypair and existing Curve25519 key followed by HKDF is applied to derived a key which is used to wrap and unwrap. The wrapped key and the ephemeral public key is sent to the user	Ed25519 Key generation Ed25519 shared secret generation HKDF RFC5869 AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
NVM Storage Controller Key Service	wrapping DEK using NVM storage controller key	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Elliptic Curve Integrated Encryption Scheme (ECIES) Encryption	Encryption	ECDH Shared Secret Computation ANSI X9.63 KDF AES-GCM	Crypto Officer
Elliptic Curve Integrated Encryption Scheme (ECIES) Decryption	Decryption	ECDH Shared Secret Computation ANSI X9.63 KDF AES-GCM	Crypto Officer
PBKDF Key Derivation	Hash-based Key Derivation	PBKDF	Crypto Officer

Name	Description	Algorithms	Role
File system DEK service	Unwrap the DEK using referenced class key and re-wrap using NVM storage controller key	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Generation of DEK via IPC using class D key	Requesting generate DEK service via IPC Channel using class D keys	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer
Requesting backup keybag service via IPC using class D key	Requesting backup keybag service via IPC Channel using class D keys	AES KW using class D key, keys from Device keybag, keys from iCloud keybag or NVM storage controller key	Crypto Officer

Table 19: Non-Approved Services

4.5 External Software/Firmware Loaded

N/A

5 Software/Firmware Security

5.1 Integrity Techniques

The Apple corecrypto Module v12.0 [Apple silicon, Secure Key Store, Hardware, SL2/PHY3] is in the form of binary executable code. A firmware 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. As the module is delivered built with the Device OS, there is no standalone delivery of the module. The vendor's internal development process guarantees that the correct version of module goes with its intended Device OS version.

5.2 Initiate on Demand

The module's integrity test can be performed on demand by powering-off and reloading the module. The integrity test on demand is performed as part of the Pre-Operational Self-Tests, automatically executed at power-on.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

6.2 Configuration Settings and Restrictions

The module operates within the sepOS execution environment which is separate from the Device OS execution environment. The SEP operating system provides memory isolation between all applications executing on it. The Device OS is unable to access the module's memory or observe the module's operation.

7 Physical Security

The defined physical boundary of the Apple corecrypto Module v12 [Apple silicon, Secure Key Store, Hardware, SL2/PHY3] is the entire System-on-Chip (SoC) listed in the Tested Module Identification table. Consequently, the physical embodiment of each SoC is considered to be that of a single-chip cryptographic module.

The hardware module conforms to the Level 3 requirements for physical security. The physical components that comprise the module are of production grade components with industry standard passivation applied. The module is covered with tamper-evident coating that deter direct observation, probing, or manipulation of the single-chip as detailed in the Physical Security Mechanisms and Actions Required table. The hardness of the coated material was tested in the module's intended temperature range of operation (Hardness Testing Temperature Ranges Table). The module correctly implements the Environmental Failure Protection (EFP) features as detailed in the EFP/EFT Information Table.

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
Production Grade Components that include standard passivation	No operator-performed testing is recommended	N/A
Tamper-Evident Coating or black hard coated material or metal coating, SoC is soldered in logic board from the Ball Grid Array (BGA) or SIP is embedded in hardened resin. The components listed above are opaque within the visible spectrum.	No operator-performed testing is recommended	N/A
Hardness of the coating	No operator-performed testing is recommended	N/A
Environmental Failure Protection (EFP) forces the module to shut down	No operator-performed testing is recommended	N/A

Table 20: Mechanisms and Actions Required

7.2 User Placed Tamper Seals

Number:
Placement:
Surface Preparation:
Operator Responsible for Securing Unused Seals:
Part Numbers:

7.3 EFP/EFT Information

Temp/Voltage Type	Temperature or Voltage	EFP or EFT	Result
LowTemperature	Values found in Apple proprietary document	EFP	shutdown
HighTemperature	Values found in Apple proprietary document	EFP	shutdown
LowVoltage	Values found in Apple proprietary document	EFP	shutdown
HighVoltage	Values found in Apple proprietary document	EFP	shutdown

Table 21: EFP/EFT Information

N/A

7.4 Hardness Testing Temperature Ranges

Temperature Type	Temperature
LowTemperature	-25 Celcius
HighTemperature	51 Celcius

Table 22: Hardness Testing Temperatures

N/A

8 Non-Invasive Security

8.1 Mitigation Techniques

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
Flash	Preloaded at factory	Static
RAM	Volatile memory	Dynamic

Table 23: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
User Input	User	RAM	Plaintext	Manual	Direct	
Export Keybag from Flash	Flash	Operating calling application (TOEPP)	Encrypted	Automated	Electronic	key wrapping / key unwrapping
Export Keybag from RAM	RAM	Operating calling application (TOEPP)	Encrypted	Automated	Electronic	key wrapping / key unwrapping
Obfuscation of User Input Authentication Credential	User	RAM	Plaintext	Manual	Direct	
Obtained from ENT (P)	ENT (P)	RAM	Plaintext	Automated	Electronic	Random Number Generation
Pre-loaded from Factory	Factory install	Flash	Plaintext	Automated	Electronic	

Table 24: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Keys and SSPs (including temporary SSPs) are zeroised when the appropriate context object is destroyed by overwriting the entire context object with all zeros. The zeroization occurs at the end of an API function that uses the CSPs or when the system is powered down or when the

"Device Wipe" service is invoked. Data output interfaces are inhibited while zeroisation is performed.

Zeroization Method	Description	Rationale	Operator Initiation
Context object destruction	SSPs are zeroised when the appropriate context object is destroyed	Zeroization when structure is deallocated	N/A
Power Down	SSPs are zeroised when the system is powered down	Powering down forces context object destruction	Operator can initiate a power down
Device Wipe	Erase all content (factory reset)	Factory reset zeroizes all SSPs, including those stored in Flash	Operator can initiate a device wipe

Table 25: SSP Zeroization Methods

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys)	AES keys in user keybag	256-bits - 256-bits	Symmetric - CSP			Unauthenticated Symmetric Encryption and Decryption key wrapping / key unwrapping
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys)	AES keys in backup keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		key wrapping / key unwrapping
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in	AES keys in escrow keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		key wrapping / key unwrapping

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Escrow Keybag (AES keys)						
Data Encryption Key (DEK) (AES key)	AES keys in user keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		key wrapping / key unwrapping Random Number Generation
Entropy input string	Entropy input string	256-bits - 256-bits	Entropy - CSP	Random Number Generation		Random Number Generation
DRBG seed	DRBG seed derived from entropy input (IG D.L compliant)	384-bits - 256-bits	Seed - CSP	Random Number Generation		Random Number Generation
DRBG internal state (V value, Key)	Internal state values associated with CTR_DRBG	384-bits - 256-bits	DRBG - CSP	Random Number Generation		Random Number Generation
HMAC key	HMAC key	112-bits - 112-bits	Message Authentication Key - CSP	Symmetric Key Generation		Random Number Generation
Authentication Credential	User-provided credentials	N/A - N/A	User-generated - CSP			key wrapping / key unwrapping

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
REK	Root Encryption Key	256-bits - 256-bits	Symmetric - CSP			key wrapping / key unwrapping

Table 26: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys)	Export Keybag from Flash Pre-loaded from Factory	Flash:Encrypted	From factory install to device-wipe	Device Wipe	
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys)	Export Keybag from RAM	RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Escrow Keybag (AES keys)	Export Keybag from RAM	RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	
Data Encryption Key (DEK) (AES key)	Export Keybag from RAM	RAM:Encrypted	From service invocation to service	Context object destruction Power Down	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
			completion		
Entropy input string	Obtained from ENT (P)	RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	DRBG seed:Derives
DRBG seed		RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	Entropy input string:Derived From DRBG internal state (V value, Key):Generates
DRBG internal state (V value, Key)		RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	DRBG seed:Generated From
HMAC key		RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	
Authentication Credential	User Input	RAM:Obfuscated	From service invocation to service completion	Context object destruction Power Down	REK:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
REK	Obfuscation of User Input Authentication Credential	RAM:Plaintext	From service invocation to service completion	Context object destruction Power Down	Authentication Credential:Obfuscation from

Table 27: 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 firmware integrity automatically when the module is loaded into memory (i.e., at power on) before the module transitions to the operational state. A firmware integrity test is performed on the firmware component of the module. The module's HMAC-SHA256 is used as an approved integrity technique. Prior to using HMAC-SHA-256, a Conditional Cryptographic Algorithm Self-Tests (CAST) KAT is performed on the HMAC algorithm.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A2845)	112-bit key	Message Authentication	SW/FW Integrity	If the test fails, then the module enters an Error State.	The HMAC value is pre-computed at build time and stored in the module. The HMAC value is recalculated during runtime and compared with the stored value.

Table 28: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC-SHA2-512 (A2845)	112-bit key	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
HMAC-SHA2-512 (A2848)	112-bit key	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-SHA2-256 (A2849)	112-bit key	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
SHA2-256 (A2845)	N/A	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
SHA2-256 (A2848)	N/A	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
SHA-1 (A2845)	N/A	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
SHA-1 (A2848)	N/A	KAT	CAST	Module becomes operational	Message authentication	Test runs at Power-on before the integrity test
AES-CBC (A2842)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CBC (A2842)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-KW (A2843)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-KW (A2843)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-CBC (A2844)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CBC (A2844)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-KW (A2845)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-KW (A2845)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-KW (A2846)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-KW (A2846)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-ECB (A2847)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-ECB (A2847)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-CBC (A510)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CBC (A510)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-ECB (A501)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-ECB (A501)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-ECB (A1362)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-ECB (A1362)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						integrity test
AES-CBC (A1469)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CBC (A1469)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-CBC (A2863)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-CBC (A2863)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
AES-ECB (A2864)	128-bit key	KAT	CAST	Module becomes operational	Encryption	Test runs at Power-on before the integrity test
AES-ECB (A2864)	128-bit key	KAT	CAST	Module becomes operational	Decryption	Test runs at Power-on before the integrity test
Counter DRBG (A1362)	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

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG (A2864)	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
Counter DRBG (A501)	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
ESV-RCT (Startup)	Repetition Count Test performed at entropy source startup	fault-detection test	CAST	successful seeding of SP 800-90A DRBG	SP 800-90B 4.4.1 Repetition Count Test	upon startup of entropy source
ESV-RCT (Continuous)	Repetition Count Test performed every invocation of entropy source after startup	fault-detection test	CAST	successful seeding of SP 800-90A DRBG	SP 800-90B 4.4.1 Repetition Count Test	upon seeding or reseeding SP 800-90A DRBG
ESV-APT (Startup)	Adaptive Proportion Test performed at entropy source startup	fault-detection test	CAST	successful seeding of SP 800-90A DRBG	SP 800-90B 4.4.2 Adaptive Proportion Test	upon startup of entropy source
ESV-APT (Continuous)	Adaptive Proportion Test performed at every invocation of entropy source every	fault-detection test	CAST	successful seeding of SP 800-90A DRBG	SP 800-90B 4.4.2 Adaptive Proportion Test	upon seeding or reseeding SP 800-90A DRBG

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
	invocation after startup					

Table 29: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A2845)	Message Authentication	SW/FW Integrity	Whenever module is powered on	Upon every power-on

Table 30: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-512 (A2845)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A2848)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A2849)	KAT	CAST	On Demand	Manually
SHA2-256 (A2845)	KAT	CAST	On Demand	Manually
SHA2-256 (A2848)	KAT	CAST	On Demand	Manually
SHA-1 (A2845)	KAT	CAST	On Demand	Manually
SHA-1 (A2848)	KAT	CAST	On Demand	Manually
AES-CBC (A2842)	KAT	CAST	On Demand	Manually
AES-CBC (A2842)	KAT	CAST	On Demand	Manually
AES-KW (A2843)	KAT	CAST	On Demand	Manually
AES-KW (A2843)	KAT	CAST	On Demand	Manually
AES-CBC (A2844)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A2844)	KAT	CAST	On Demand	Manually
AES-KW (A2845)	KAT	CAST	On Demand	Manually
AES-KW (A2845)	KAT	CAST	On Demand	Manually
AES-KW (A2846)	KAT	CAST	On Demand	Manually
AES-KW (A2846)	KAT	CAST	On Demand	Manually
AES-ECB (A2847)	KAT	CAST	On Demand	Manually
AES-ECB (A2847)	KAT	CAST	On Demand	Manually
AES-CBC (A510)	KAT	CAST	On Demand	Manually
AES-CBC (A510)	KAT	CAST	On Demand	Manually
AES-ECB (A501)	KAT	CAST	On Demand	Manually
AES-ECB (A501)	KAT	CAST	On Demand	Manually
AES-ECB (A1362)	KAT	CAST	On Demand	Manually
AES-ECB (A1362)	KAT	CAST	On Demand	Manually
AES-CBC (A1469)	KAT	CAST	On Demand	Manually
AES-CBC (A1469)	KAT	CAST	On Demand	Manually
AES-CBC (A2863)	KAT	CAST	On Demand	Manually
AES-CBC (A2863)	KAT	CAST	On Demand	Manually
AES-ECB (A2864)	KAT	CAST	On Demand	Manually
AES-ECB (A2864)	KAT	CAST	On Demand	Manually
Counter DRBG (A1362)	KAT	CAST	On Demand	Manually
Counter DRBG (A2864)	KAT	CAST	On Demand	Manually
Counter DRBG (A501)	KAT	CAST	On Demand	Manually
ESV-RCT (Startup)	fault-detection test	CAST	On Demand	Manually
ESV-RCT (Continuous)	fault-detection test	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
ESV-APT (Startup)	fault-detection test	CAST	On Demand	Manually
ESV-APT (Continuous)	fault-detection test	CAST	On Demand	Manually

Table 31: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error state	The HMAC-SHA-256 value computed over the module did not match the pre- computed value, OR the computed value in the invoked Conditional CAST did not match the known value. No cryptographic services are provided, and data output is prohibited	Pre-operational Firmware Integrity Test failure OR Conditional CAST failure	Power off/on	for Integrity: print statement "FAILED: fipspost_post_integrity" to stdout; for CAST: sprint statement "FAILED:<event>" to stdout (<event> refers to any of the cryptographic functions listed in the Conditional Self-test Table)

Table 32: 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 reloading the module.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Startup Procedures: As the module is delivered built with the Device OS, there is no standalone delivery of the module.

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 used 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 biometric authentication option provided by the underlying test platform shall be disabled in order to run the module in the FIPS validated manner.

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/projects/cryptographic-module-validation-program/entropy-validations/certificate/113

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

The User role is authenticated with the mechanism described in section 4. The User role can access the module via mailbox interface using the Device OS's XNU kernel. The User role can perform subset of services from Table - Approved Algorithms.

As stated in the Administrator Guidance section above, 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. This transition cannot be made by the User directly, as all non-approved services require an implicit transition into the Crypto-Officer role. Any calling of such services is therefore implicitly performed by the Crypto Officer.

11.4 End of Life

The Device Wipe service erases the module content. When performing a Device Wipe service to erase all content of the module, the procedure must be performed under the control of the Operator.

12 Mitigation of Other Attacks

The module does not claim mitigation of other attacks.

Appendix A. Glossary and Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interfaces
APT	Adaptive Proportion Test (SP800-90B health test)
BGA	Ball Grid Array (Physical Security)
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CMVP	Cryptographic Module Validation Program
CST	Cryptographic and Security Testing
CTR	Counter Mode
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECDSA	DSA (Digital Signature Algorithm) based on Elliptic Curve Cryptography (ECC)
EMI	Electromagnetic Interference (Physical Security)
ESV	NIST entropy source validation program providing SP 800-90B compliant entropy validation certificate
FIPS	Federal Information Processing Standards Publication
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
IPC	Inter-Process Communication
IHS	Integrated Heat Spreader (Physical Security)
KAT	Known Answer Test
KDF	Key Derivation Function
KEK	Key Encryption Key
KW	AES Key Wrap
MAC	Message Authentication Code
NIST	National Institute of Science and Technology
NVM	Non-Volatile Memory
OFB	Output Feedback
OS	Operating System
PBKDF	Password Based Key Derivation Function
RCT	Repetition Count Test (SP800-90B health test)
SEP	Secure Enclave Processor
SHA	Secure Hash Algorithm

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 January 2024 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS%20140-3%20IG.pdf
FIPS140-3_MM	CMVP FIPS 140-3 Management Manual February 2024 https://csrc.nist.gov/csrc/media/Projects/cryptographic-module-validation-program/documents/fips%20140-3/FIPS-140-3-CMVP%20Management%20Manual%20v2.1%5B02-29-2024%5D.pdf
SP 800-140	FIPS 140-3 Derived Test Requirements (DTR) March 2020 https://csrc.nist.gov/publications/detail/sp/800-140/final
SP 800-140A	CMVP Documentation Requirements March 2020 https://csrc.nist.gov/publications/detail/sp/800-140a/final
SP 800-140Br1	CMVP Security Policy Requirements November 2023 https://doi.org/10.6028/NIST.SP.800-140Br1
SP 800-140C	CMVP Approved Security Functions July 2023 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-140Cr2.pdf
SP 800-140D	CMVP Approved Sensitive Security Parameter Generation and Establishment Methods July 2023 https://doi.org/10.6028/NIST.SP.800-140Dr2
SP 800-140E	CMVP Approved Authentication Mechanisms March 2020 https://csrc.nist.gov/publications/detail/sp/800-140e/final
SP 800-140F	CMVP Approved Non-Invasive Attack Mitigation Test Metrics March 2020 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
FIPS186-5	**Digital Signature Standard (DSS)** F3b 2023 https://doi.org/10.6028/NIST.FIPS.186-5
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
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
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
SP800-38C	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

SP800-56Cr2	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 800-108r1 - Recommendation for Key Derivation Using Pseudorandom Functions Aug 2022 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
SP800-133r2	**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 Platform Certifications https://support.apple.com/guide/certifications/welcome/web