Apple Inc.

Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2]

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

© 2025 Apple Inc., All rights reserved.
This document may be reproduced and distributed only in its original entirely without revision.

Table of Contents

1 General 6
1.1 Overview 6
1.2 Security Levels 6
2 Cryptographic Module Specification 7
2.1 Description 7
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.3 Excluded Components 10
2.4 Modes of Operation 10
2.5 Algorithms 10
2.6 Security Function Implementations 14
2.7 Algorithm Specific Information 16
2.8 RBG and Entropy 16
2.9 Key Generation 16
2.10 Key Establishment 16
2.11 Industry Protocols 16
3 Cryptographic Module Interfaces 17
3.1 Ports and Interfaces 17
4 Roles, Services, and Authentication 18
4.1 Authentication Methods 18
4.2 Roles 19
4.3 Approved Services 19
4.4 Non-Approved Services 27
4.5 External Software/Firmware Loaded 29
5 Software/Firmware Security 30
5.1 Integrity Techniques 30
5.2 Initiate on Demand 30
6 Operational Environment 31
6.1 Operational Environment Type and Requirements 31
6.2 Configuration Settings and Restrictions 31
7 Physical Security 32
7.1 Mechanisms and Actions Required 32
8 Non-Invasive Security 33
8.1 Mitigation Techniques 33
9 Sensitive Security Parameters Management 34

© 2025 Apple Inc., All rights reserved.
This document may be reproduced and distributed only in its original entirely without revision.

9.1 Storage Areas 34
9.2 SSP Input-Output Methods 34
9.3 SSP Zeroization Methods 34
9.4 SSPs 35
10 Self-Tests 39
10.1 Pre-Operational Self-Tests 39
10.2 Conditional Self-Tests 39
10.3 Periodic Self-Test Information 40
10.4 Error States 41
10.5 Operator Initiation of Self-Tests 42
11 Life-Cycle Assurance 43
11.1 Installation, Initialization, and Startup Procedures 43
11.2 Administrator Guidance 43
11.3 Non-Administrator Guidance 43
11.4 End of Life 44
12 Mitigation of Other Attacks 45

© 2025 Apple Inc., All rights reserved.
This document may be reproduced and distributed only in its original entirely without revision.

List of Tables

Table 1: Security Levels 6
Table 2: Tested Module Identification – Hardware 9
Table 3: Modes List and Description 10
Table 4: Approved Algorithms 13
Table 5: Vendor-Affirmed Algorithms 13
Table 6: Non-Approved, Not Allowed Algorithms 14
Table 7: Security Function Implementations 15
Table 8: Entropy Certificates 16
Table 9: Entropy Sources 16
Table 10: Ports and Interfaces 17
Table 11: Authentication Methods 18
Table 12: Roles 19
Table 13: Approved Services 26
Table 14: Non-Approved Services 29
Table 15: Mechanisms and Actions Required 32
Table 16: Storage Areas 34
Table 17: SSP Input-Output Methods 34
Table 18: SSP Zeroization Methods 35
Table 19: SSP Table 1 36
Table 20: SSP Table 2 38
Table 21: Pre-Operational Self-Tests 39
Table 22: Conditional Self-Tests 40
Table 23: Pre-Operational Periodic Information 41
Table 24: Conditional Periodic Information 41
Table 25: Error States 41

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

List of Figures

Figure 1: Block Diagram 8
Figure 2: Apple A10 Fusion 9
Figure 3: Apple A10X Fusion 9
Figure 4: Apple A12 Bionic 9
Figure 5: Apple A12Z Bionic 9
Figure 6: Apple T Series T2 9
Figure 7: Apple S6 9
Figure 8: Apple S7 9
Figure 9: Apple S8 9

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] 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 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	2
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

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:
The Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] 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' (iOS 17, iPadOS 17, watchOS 10, tvOS 17, T2OS 14) 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: Single Chip

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 shared memory segment (called Mailbox) that can be accessed by both SKS and the Device OS's XNU kernel, supported with an interrupt system and used by XNU to request services of the SKS module.
- A volatile RAM for storing runtime SSPs.
- A non-volatile Flash for storing an encrypted Class D key.

© 2025 Apple Inc., All rights reserved.
This document may be reproduced and distributed only in its original entirely without revision.

The physical perimeter is represented by the exterior-most red line in the block diagram below.

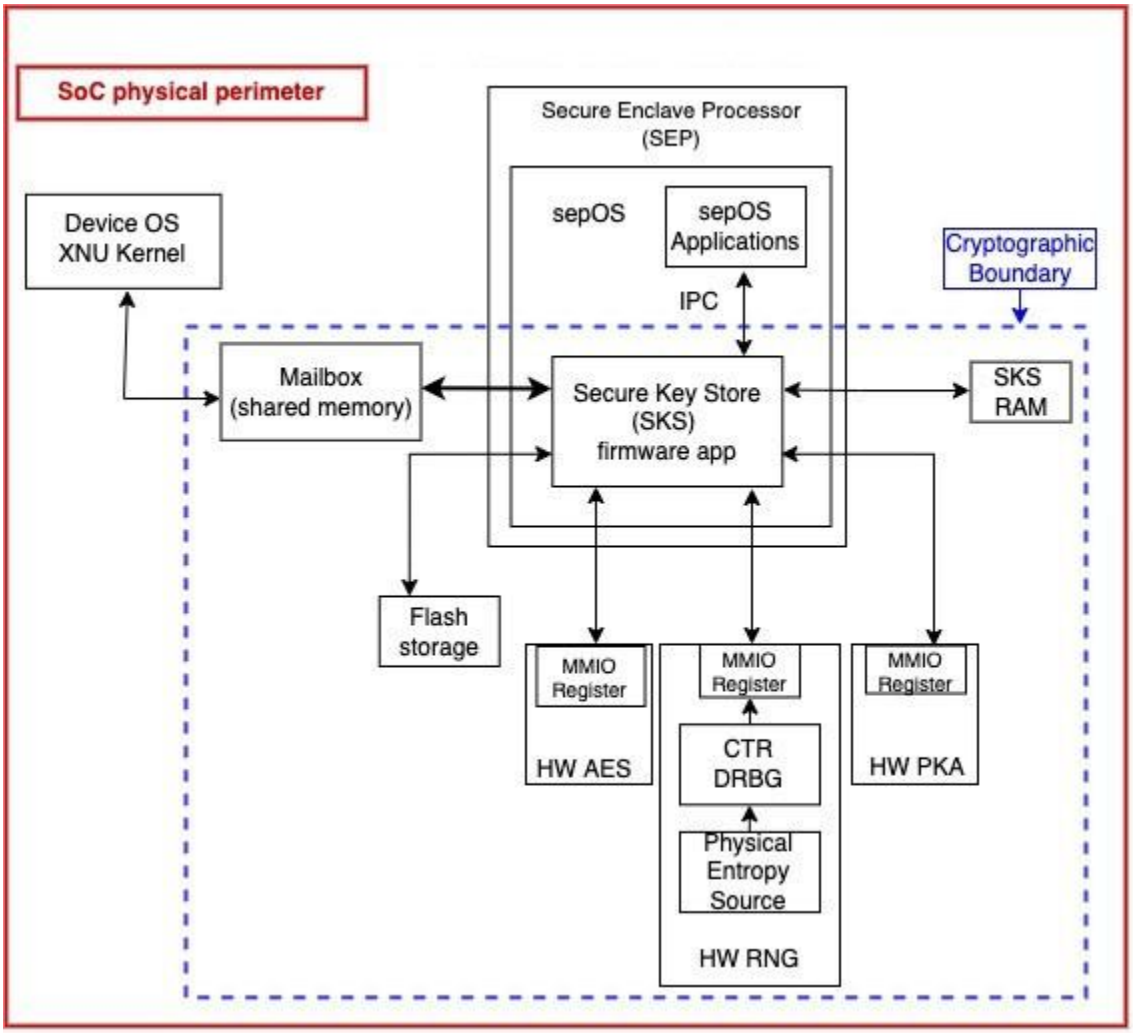

Figure 1: Block Diagram

© 2025 Apple Inc., All rights reserved.
This document may be reproduced and distributed only in its original entirely without revision.

Tested Operational Environment's Physical Perimeter (TOEPP):

The photograph of each hardware module is shown below:

Figure 2: Apple A10 Fusion

Figure 3: Apple A10X Fusion

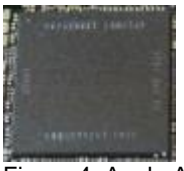
Figure 4: Apple A12 Bionic

Figure 5: Apple A12Z Bionic

Figure 6: Apple T Series T2

Figure 7: Apple S6

Figure 8: Apple S7

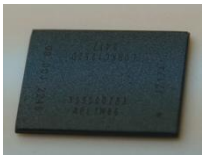
Figure 9: Apple S8

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
sepOS distributed with iPadOS 17 running on iPad (7th generation)	2.0	14.0	Apple A Series A10 Fusion	N/A
sepOS distributed with iPadOS 17 running on iPad Pro 10.5 inch	2.0	14.0	Apple A Series A10X Fusion	N/A
sepOS distributed with iPadOS 17 running on iPad mini (5th generation)	2.0	14.0	Apple A Series A12 Bionic	N/A
sepOS distributed with iPadOS 17 running on iPad Pro 11-inch (2nd generation)	2.0	14.0	Apple A Series A12Z Bionic	N/A
sepOS distributed with tvOS 17 running on Apple TV 4K (2nd generation)	2.0	14.0	Apple A Series A12 Bionic	N/A
sepOS distributed with watchOS 10 running on Apple Watch Series S6	2.0	14.0	Apple S Series S6	N/A
sepOS distributed with watchOS 10 running on Apple Watch Series S7	2.0	14.0	Apple S Series S7	N/A
sepOS distributed with watchOS 10 running on Apple Watch Series S8	2.0	14.0	Apple S Series S8	N/A
sepOS distributed with T2OS 14 running on Apple Security Chip T2	2.0	14.0	Apple T Series T2	N/A

Table 2: Tested Module Identification – Hardware

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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	Automatically entered whenever an approved service is requested	Approved	The return of a '0' from fips_allowed_mode() for block cipher functions and fips_allowed() for all other services indicates the executed cryptographic algorithm was approved
Non-approved mode	Automatically entered whenever a non-approved service is requested	Non-Approved	The return of any non-zero value from fips_allowed_mode() for block cipher functions and fips_allowed() for all other services indicates the executed cryptographic algorithm was non-approved

Table 3: Modes List and Description

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A1469	Direction - Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A2863	Direction - Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A5975	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A5976	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A5977	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	A5978	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CBC	C317	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CBC	C318	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CBC	C322	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-CBC	C330	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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	A2863	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A2864	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	A3490	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	A501	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	A5975	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5976	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5978	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A5980	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	C317	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	C318	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	C322	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-ECB	C323	Direction - Encrypt Key Length - 256	SP 800-38A
AES-ECB	C330	Direction - Decrypt, Encrypt Key Length - 128, 256	SP 800-38A
AES-KW	A5976	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A5978	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
AES-KW	A5979	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38F
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	A3490	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A501	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	C323	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	DRBG 2023	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	DRBG 2024	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
Counter DRBG	DRBG 2029	Prediction Resistance - Yes Mode - AES-256 Derivation Function Enabled - No	SP 800-90A Rev. 1
HMAC-SHA-1	A5978	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA-1	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5978	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-224	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5978	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-256	A5982	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5978	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-384	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5978	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
HMAC-SHA2-512/256	A5981	Key Length - Key Length: 8-262144 Increment 8	FIPS 198-1
SHA-1	A5978	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA-1	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-224	A5978	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-224	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Algorithm	CAVP Cert	Properties	Reference
SHA2-256	A5978	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-256	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-256	A5982	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-384	A5978	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-384	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512	A5978	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4
SHA2-512/256	A5981	Message Length - Message Length: 0-32768 Increment 8	FIPS 180-4

Table 4: Approved Algorithms

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Symmetric AES key:Key Length/ Key Strength: 256	N/A	Symmetric Key Generation (CKG using method in Section 4 example 1 [SP 800-133Rev2])

Table 5: Vendor-Affirmed Algorithms

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

Name	Use and Function
Ed25519 Key generation	EdDSA signature scheme
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

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Use and Function
ECDH Shared Secret Computation	Shared secret computation using P curves.
ANSI X9.63 KDF	Hash based KDF
AES-GCM	Encryption and Decryption
HKDF RFC5869	HMAC based Key Derivation Function
PBKDF	Key Derivation
ECDSA implemented in FW	Key generation as part of Ref key generation service and validation, Signature generation and verification using Ref-key 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 using Ref-key
ECDH implemented in HW PKA	Shared secret computation using Ref-key
AES KW	Key wrapping and unwrapping using class D key, keys from Device keybag, keys from iCloud keybag, keys from Escrow keybag, keys from any keybag used with Class B Curve 25519 encrypt/decrypt, keys from Backup keybag used for wrapping Ed25519 keys, or NVM storage controller key

Table 6: Non-Approved, Not Allowed Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Symmetric Encryption and Decryption	BC-UnAuth	Symmetric Encryption and Decryption	AES-CBC:Key Length / Key Strength:128, 192, 256 AES-ECB:Key Length / Key Strength:256	AES-CBC: (A1469, A2863, A5975, A5976, A5977, A5978, C317, C318, C322, C330) AES-ECB: (A1362, A1469, A2863, A2864, A3490, A501, A5975, A5976, A5978, A5980, C317, C318, C322, C323, C330)
Key Wrapping	KTS-Wrap BC-Auth	Key Wrapping	AES-KW:Key Length / Key	AES-KW: (A5976, A5978, A5979)

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Type	Description	Properties	Algorithms
			Strength:128, 192, 256	
Random Number Generation	DRBG	Random Number Generation	Counter DRBG:AES-256; No Derivation Function; Prediction Resistance Enabled; Key Length / Key Strength: 256	Counter DRBG: (A1362, A2864, A3490, A501, C323, DRBG 2023, DRBG 2024, DRBG 2029)
Keyed Hash	MAC	Keyed Hash	HMAC-SHA-1:112 bits or greater HMAC-SHA2-224:112 bits or greater HMAC-SHA2-256:112 bits or greater HMAC-SHA2-384:112 bits or greater HMAC-SHA2-512:112 bits or greater HMAC-SHA2-512/256:112 bits or greater	HMAC-SHA-1: (A5978, A5981) HMAC-SHA2-224: (A5978, A5981) HMAC-SHA2-256: (A5978, A5981, A5982) HMAC-SHA2-384: (A5978, A5981) HMAC-SHA2-512: (A5978, A5981) HMAC-SHA2-512/256: (A5981)
Message Digest	SHA	Message Digest	SHA-1:N/A SHA2-224:N/A SHA2-256:N/A SHA2-384:N/A SHA2-512:N/A SHA2-512/256:N/A	SHA-1: (A5978, A5981) SHA2-224: (A5978, A5981) SHA2-256: (A5978, A5981, A5982) SHA2-384: (A5978, A5981) SHA2-512: (A5978, A5981) SHA2-512/256: (A5981)
Symmetric Key Generation	CKG	AES Key	CKG [SP800-133Rev2]:Key Length/ Key Strength: 256	CKG: ()

Table 7: Security Function Implementations

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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, 2031.

2.8 RBG and Entropy

Cert Number	Vendor Name
E113	Apple Inc.

Table 8: Entropy Certificates

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

Table 9: Entropy Sources

Entropy sources : The internal physical noise source consists 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

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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 API offered by the module's mailbox interface to callers from the Device OS' kernel and through the 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 10: 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

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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 11: Authentication Methods

Within the constraints of FIPS 140-3 level 2, 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 AES-KW 256-bit key unwrapping function provides 256 bits of strength. Therefore, the strength of the authentication mechanism in use is 1/ 2^256. Even using a rate of 1µs per failed authentication, which would allow 60,000,000 consecutive attempts per minute (60s / 0.000001s), only provides a probability of successfully authenticating that is less than or equal to 60,000,000 * 1 / 2^256.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

All authentication data is obscured during data entry.

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 12: Roles

4.3 Approved Services

The module has an approved and non-approved mode of operation. The approved mode of operation is assumed automatically without any specific configuration. If the device starts up successfully then the module has passed all self-tests and is operating in the approved mode. Any calls to the non-approved security functions listed in the Non-Approved Services Table will cause the module to assume the non-approved mode of operation.

The module implements a dedicated API function to indicate if a requested service utilizes an approved security function. The approved service indicator was observed to utilize one of two functions (fips_allowed and fips_allowed_mode) depending on the service in question. Calling fips_allowed_mode with AES-ECB, AES-CBC or AES-KW will returned a zero to indicate it was an approved algorithm. Similarly, calling fips_allowed with any other approved algorithm returned zero. Calling either of these with an algorithm not listed in the Approved Algorithms Table returned a non-zero value, and as such indicated a non-approved service.

The table below lists all approved services that can be used in the approved mode of operation by authorized operators of either the User or Crypto Officer Roles. The abbreviations of the access rights to keys and 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

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	Success returned from API	User credential, reference to	status (success/error)	Symmetric Encryption and Decrypt	User - Class A, Class C, Class AK, Class

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	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 mailbox interface and the REK is zeroized.	listed in the customer proprietary guidance document	class C/A key from the user keybag		ion Key Wrapping	AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - REK: W,E - Authentication Credential: W,E
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	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - REK: W,E - Authentication

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 sent out of the module	Success returned from API listed in the customer proprietary guidance document	reference to class C/A key from the User keybag	wrapped DEK	Key Wrapping Random Number Generation 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 - 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	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	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E - Data Encryption Key (DEK)

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	error is returned; Step 3. plaintext DEK is sent out to the User. (AS09.16)					(AES key): R,E
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 Symmetric Key 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 internal state: V value, key, and seed material: 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	wrapped DEK, reference to class C or A key from the user keybag	wrapped DEK	Key Wrapping	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in User Keybag (AES keys): W,E

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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.	document				- Data Encryption Key (DEK) (AES key): W,E - Entropy input string: W,E - DRBG internal state: V value, key, and seed material: W,E - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys): G,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 Symmetric Key Generation	User - Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Escrow Keybag (AES keys): G,E - Entropy

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						input string: W,E - DRBG internal state: V value, key, and seed material: 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	Keyed Hash 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 in

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						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 Encryption Key (DEK)

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						(AES key): Z - Entropy input string: Z - DRBG internal state: V value, key, and seed material: Z - HMAC key: Z - Authentication Credential: Z - REK: 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	Symmetric Encryption and Decryption Key Wrapping Random Number Generation Keyed Hash Message Digest	Crypto Officer
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 13: Approved Services

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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	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	Crypto Officer
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	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	Crypto Officer
Generate Ref-Key	Key Generation	ECDSA implemented in FW ECDSA implemented in HW PKA	Crypto Officer
Sign and verify using Ref-key	Signature Generation and Verification	ECDSA implemented in FW	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	Curve 25519 shared secret generation ECDH Shared Secret Computation AES-GCM HKDF RFC5869 AES KW	Crypto Officer
Generate Shared Secret using Ref-key	Shared secret generation	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	Crypto Officer

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Algorithms	Role
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	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	Crypto Officer
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	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	Crypto Officer
Wrap or unwrap service for keychain using DK/DKU/CK/ CKU/AK/AKU/AKPU Ed25519 key from asymmetric keybag	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 HKDF RFC5869 AES KW	Crypto Officer
Asymmetric (Ed25519) backup keybag wrap and unwrap	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 HKDF RFC5869 AES KW	Crypto Officer
NVM Storage Controller Key Service	wrapping DEK using NVM storage controller key	AES KW	Crypto Officer
Elliptic Curve Integrated Encryption Scheme (ECIES) Encryption	Encryption	ECDH Key Pair Generation ANSI X9.63 KDF AES-GCM	Crypto Officer
Elliptic Curve Integrated Encryption	Decryption	ECDH Key Pair Generation	Crypto Officer

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Algorithms	Role
Scheme (ECIES) Decryption		ANSI X9.63 KDF AES-GCM	
PBKDF Key Derivation	Hash-based Key Derivation	PBKDF	Crypto Officer
File system DEK service	Unwrap the DEK using referenced class key and re-wrap using NVM storage controller key	AES KW	Crypto Officer
Generation of DEK via IPC using class D key	Requesting generate DEK service via IPC Channel using class D keys	AES KW	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	Crypto Officer

Table 14: Non-Approved Services

4.5 External Software/Firmware Loaded

N/A

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

5 Software/Firmware Security

5.1 Integrity Techniques

The Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] 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.

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

7 Physical Security

The defined physical boundary of the Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] is the entire System-on-Chip (SoC) listed in the Tested Module Identification - Hardware 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 2 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.

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

Table 15: Mechanisms and Actions Required

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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 16: Storage Areas

During runtime operation, the Apple corecrypto Module v14.0 [Apple silicon, Secure Key Store, Hardware, SL2] module stores keys/SSPs in volatile memory, except for the user keybag that is stored in Flash. The module protects all keys/SSPs through the memory separation and protection mechanisms provided by the operating system while the Flash component only provides exclusive access to the module. No process other than the module itself can access the keys/SSPs in its process memory or Flash component.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Mailbox Input	User	RAM	Plaintext	Manual	Electronic	
Export DEK	RAM	Operating calling application (TOEPP)	Plaintext	Automated	Electronic	
Export Key or Keybag from Flash	Flash	Operating calling application (TOEPP)	Encrypted	Automated	Electronic	Key Wrapping
Export Key or Keybag from RAM	RAM	Operating calling application (TOEPP)	Encrypted	Automated	Electronic	Key Wrapping

Table 17: SSP Input-Output Methods

Per the definition in IG 2.3.B, "*Transferring SSPs including the entropy input between a sub-chip cryptographic subsystem and an intervening functional subsystem for Security Levels 1 and 2 on the same single chip is considered as not having Sensitive Security Parameter Establishment crossing the HMI*". As such, the import or export Keys/SSP as defined in Table 1 of IG 9.5.A do not apply.

Within the TOEPP, keys and SSPs can either be entered, or output from the Apple Secure Key Store Cryptographic Module to/from intervening functional subsystems in plaintext .

9.3 SSP Zeroization Methods

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Zeroization Method	Description	Rationale	Operator Initiation
Context object destruction	SSPs are zeroised when the appropriate context object is destroyed. The module provides an implicit indication that the zeroisation has successfully completed by returning access to the User.	Zeroization when structure is deallocated	Operator can initiate context object destruction
Power Down	SSPs are zeroised when the system is powered down. The module provides an implicit indication that the zeroisation has successfully completed by returning access to the User.	Powering down forces context object destruction	Operator can initiate a power down
Device Wipe	Erase all content (factory reset). The module provides an implicit indication that the zeroisation has successfully completed by returning access to the User. This performs end of life of the device.	Factory reset zeroizes all SSPs, including those stored in Flash	Operator can initiate a device wipe

Table 18: SSP Zeroization Methods

Input and output interfaces are inhibited while zeroization is performed. Zeroization is immediate and uninterruptible, preventing the retrieval and reuse of the zeroized values. The module provides an implicit indication that the zeroization has successfully completed by returning access to the User, ready to service the next request.

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			Symmetric Encryption and Decryption Key Wrapping
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup	AES keys in backup keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		Key Wrapping

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Keybag (AES keys)						
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Escrow Keybag (AES keys)	AES keys in escrow keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		Key Wrapping
Data Encryption Key (DEK) (AES key)	AES keys in user keybag	256-bits - 256-bits	Symmetric - CSP	Symmetric Key Generation		Symmetric Encryption and Decryption
Entropy input string	Entropy input string	256-bits - 256-bits	Entropy - CSP			Random Number Generation
DRBG internal state: V value, key, and seed material	Internal state values associated with CTR_DRBG	256-bits - 256-bits	DRBG - CSP	Random Number Generation		Random Number Generation
HMAC key	HMAC key	112-bits - 112-bits	Message Authentication Key - CSP			Keyed Hash
Authentication Credential	User-provided credentials	N/A - N/A	User-generated - CSP			Key Wrapping
REK	Root Encryption Key	256-bits - 256-bits	Symmetric - CSP			Key Wrapping

Table 19: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Class A, Class C, Class AK, Class AKU, Class CK,	Export Key or Keybag from Flash	Flash:Encrypted	From factory install to device-wipe	Device Wipe	

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Class CKU in User Keybag (AES keys)					
Class A, Class C, Class AK, Class AKU, Class CK, Class CKU in Backup Keybag (AES keys)	Export Key or 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 Key or Keybag from RAM	RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	
Data Encryption Key (DEK) (AES key)	Export DEK Export Key or Keybag from RAM	RAM:Obfuscated	From service invocation to service completion	Context object destruction Power Down	DRBG internal state: V value, key, and seed material:Derived From
Entropy input string		RAM:Plaintext	From service invocation to service completion	Context object destruction Power Down	DRBG internal state: V value, key, and seed material:Used With
DRBG internal state: V value, key, and seed material		RAM:Plaintext	From service invocation to service completion	Context object destruction Power Down	Entropy input string:Derived From
HMAC key	Mailbox Input	RAM:Encrypted	From service invocation to service completion	Context object destruction Power Down	

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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

Table 20: SSP Table 2

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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	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 21: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Counter DRBG	128-bit key	KAT	CAST	Module becomes operational	Compliant with SP 800-90Ar1	Test runs at power-on before the integrity test
HMAC-SHA2-256	SHA2-256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA-1	SHA-1	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512	SHA2-512	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on before the integrity test
HMAC-SHA2-512/256	SHA2-512/256	KAT	CAST	Module becomes operational	Message authentication	Test runs at power-on

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
						before the integrity test
AES-CBC	128-bit key encrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-ECB	128-bit key decrypt	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-KW wrapping	wrap	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
AES-KW unwrapping	unwrap	KAT	CAST	Module becomes operational	Symmetric operation	Test runs at power-on before the integrity test
ESV-RCT Startup	Startup test with 1024 8-bit samples	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-RCT Continuous	Continuous test; Cutoff value = 31	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	Startup test with 1024 8-bit samples	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
ESV-APT Continuous	Continuous test; Cutoff value = 325	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

Table 22: Conditional Self-Tests

10.3 Periodic Self-Test Information

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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

Table 23: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG	KAT	CAST	On Demand	Manually
HMAC-SHA2-256	KAT	CAST	On Demand	Manually
HMAC-SHA-1	KAT	CAST	On Demand	Manually
HMAC-SHA2-512	KAT	CAST	On Demand	Manually
HMAC-SHA2-512/256	KAT	CAST	On Demand	Manually
AES-CBC	KAT	CAST	On Demand	Manually
AES-ECB	KAT	CAST	On Demand	Manually
AES-KW wrapping	KAT	CAST	On Demand	Manually
AES-KW unwrapping	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
ESV-APT Startup	fault-detection test	CAST	On demand	Manually
ESV-APT Continuous	fault-detection test	CAST	On demand	Manually

Table 24: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error state	No cryptographic services are provided, and data output is prohibited	Integrity test failure; CAST failure; The SP 800-90B continuous health test or startup health test 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 25: Error States

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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 module maintains separate contexts for each cryptographic operation, and there can be no CSPs used in the Non-Approved Mode of operation.

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

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.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.

12 Mitigation of Other Attacks

The module does not claim mitigation of other attacks.

© 2025 Apple Inc., All rights reserved.

This document may be reproduced and distributed only in its original entirely without revision.