Apple macOS 26 Tahoe: FileVault Security Target

Version:	1.1
Status:	Final
Date:	2026-06-29
Validation ID:	11696
Classification:	Public
Prepared for:	Apple Inc.
Prepared by:	atsec information security corporation

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.

Copyright

Legal Notice

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2026-02-20	atsec	First version
1.1	2026-06-29	atsec	Editorial update & address vendor comments. Added TD1042, TD1046.

Table of Contents

List of Tables

1 Introduction

1.1 Security Target Identification

Title:	Apple macOS 26 Tahoe: FileVault Security Target
Version:	1.1
Status:	Final
Date:	2026-06-29
Sponsor:	Apple Inc.
Developer:	Apple Inc.
Validation Body:	NIAP
Validation ID:	11696
Keywords:	FileVault, Full Drive Encryption, Encryption Engine, Authorization Acquisition

1.2 TOE Identification

The TOE is Apple macOS 26 Tahoe: FileVault.

1.3 TOE Type

The TOE type is Full Drive Encryption (Authorization Acquisition and Encryption Engine).

1.4 TOE Overview

The TOE is Apple macOS 26 Tahoe: FileVault which is a Full Drive Encryption (FDE) solution including both Authorization Acquisition (AA) and Encryption Engine (EE) components. The TOE is a built-in security feature providing data-at-rest protection on Apple Mac computers with Apple silicon. It is a hybrid FDE implementation based on a single vendor's combination of hardware and software.

The Mac computers run Apple macOS operating system. The TOE is part of the macOS operating system, which leverages the Apple silicon System on Chip (SoC). Included in the Apple silicon SoC are Apple Secure Enclave and Direct Memory Access (DMA) Storage Controller. The Secure Enclave is a dedicated secure subsystem where all FDE cryptographic key handling occurs. The DMA Storage Controller provides a dedicated AES-XTS crypto engine built into the DMA path between the storage and main memory, making data encryption with AES-XTS efficient. A special channel from the Secure Enclave securely transfers necessary keying material to the AES-XTS engine.

The macOS version used in evaluation testing is 26.3.

1.5 TOE Description

This section provides a general description of the TOE, including architecture, physical boundaries, security functions, and relevant TOE documentation and references.

1.5.1 Architecture

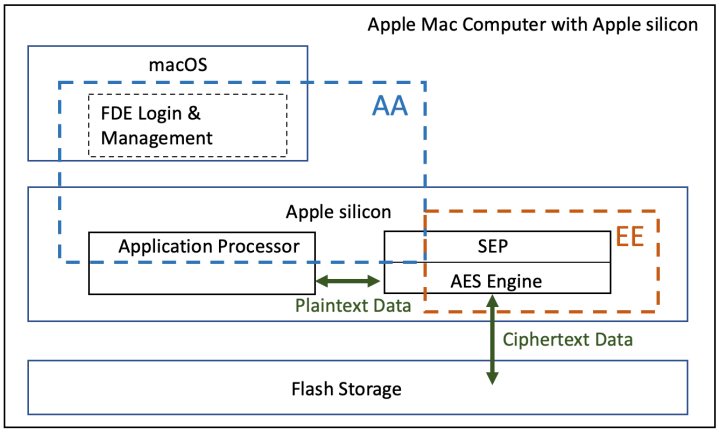
1.5.1.1 Apple silicon

The Apple silicon SoC includes:

- the application processor, which is the main processor of the TOE device and runs the macOS operating system;
- the Secure Enclave, which contains the Secure Enclave Processor (SEP) running the sepOS operating system; and
- the DMA Storage Controller, which performs the storage encryption.

The EE component is instantiated in the Secure Enclave and the DMA Storage Controller. The AA component is instantiated in the application processor (Login Window) and the Secure Enclave. The Secure Enclave provides security related functionality for EE (other than encryption/decryption of storage data), as well as all of the cryptographic functionality for AA (i.e., PBKDF2 and AES-CBC). The DMA Storage Controller provides a dedicated AES-XTS crypto engine built into the DMA path between storage and main memory of the host platform. The Login Window of the AA component is implemented as the pre-boot component on the storage drive. It captures the user password and passes it to the Secure Enclave.

Figure 1: Major Components of the TOE



1.5.1.2 Secure Enclave

The Secure Enclave is a dedicated secure subsystem integrated into the Apple silicon SoC. It is isolated from the application processor to provide an extra layer of security and is designed to keep sensitive user data secure even when the application processor's kernel becomes compromised.

The Secure Enclave contains the SEP, which runs sepOS. sepOS is bundled with macOS. The Secure Enclave also includes a hardware True Random Number Generator (TRNG) and a hardware AES-CBC engine. The TRNG and AES-CBC engine are directly connected to the SEP and are only accessible through the SEP.

Each Secure Enclave is provisioned during fabrication with its own 256-bit Unique ID (UID). This UID is used as a key by the TOE device, not accessible by other parts of the system, and not known to Apple.

1.5.2 TOE Physical Boundary

The TOE includes both hardware and software running on the Mac computers listed in Table 10: Hardware Platforms.

1.5.3 TOE Security Functionality

The TOE provides the security functions required by the conformance claims defined in Section 2.

1.5.3.1 Cryptographic Support (FCS)

The TOE includes version 26 of the Apple corecrypto cryptographic implementations and is supported by the onboard Apple Secure Enclave hardware for performing user space, kernel space, and Secure Enclave cryptographic operations. In addition, it uses a hardware-based noise source for entropy generation. The

Apple DMA Storage Controller hardware implements the AES-XTS cryptographic algorithm supported for data-at-rest encryption.

1.5.3.2 User Data Protection (FDP)

The TOE encrypts all user data using AES-XTS-256.

1.5.3.3 Security Management (FMT)

The TOE can perform management functions. The administrator has full access to carry out all management functions, whereas the user has limited privileges.

1.5.3.4 Protection of the TSF

The TOE implements the following protection of TOE Security Functionality (TSF) data:

- Protection of keys and key material.
- Compliant power saving states that purge security-relevant data upon entry.
- Cryptographic self-tests at start-up of the TOE.
- Trusted software updates using digital signatures.

The macOS operating system retrieves the update package from the Apple Update Server and forwards the package to the AA component. The TOE validates the digital certificate for the package before it is installed.

1.5.4 TOE Guidance

The following TOE guidance document in PDF format is publicly available on the NIAP Product Compliance List (PCL) alongside this Security Target:

- Apple macOS 26 Tahoe: FileVault Common Criteria Guide (CCGUIDE)

1.5.5 TOE Operational Environment

The following environmental components interoperate with the TOE in the evaluated configuration.

Table 1: TOE Operational Environment

Component	Description
Hardware platform	See Table 10: Hardware Platforms
Apple Update Server	Server that allows the TOE to download updates

2 CC Conformance Claim

This Security Target (ST) is CC Part 2 extended and CC Part 3 conformant. Common Criteria (CC) version 3.1 revision 5 is the basis for this conformance claim.

This ST claims exact conformance to the following Protection Profiles (PPs):

- PP-Configuration for Full Drive Encryption - Authorization Acquisition and Full Drive Encryption - Encryption Engine, Version 1.0 (CFG_CPP_FDE_AA-CPP_FDE_EE_V1.0).

  This PP-Configuration includes the following components:

  - collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, Version 2.0 + Errata 20190201 (CPP_FDE_AA_V2.0E);
  - collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0 + Errata 20190201 (CPP_FDE_EE_V2.0E).

The following table contains the NIAP Technical Decisions (TDs) for the collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, Version 2.0E (CPP_FDE_AA_V2.0E) at the time of the evaluation and a statement of applicability to the evaluation.

Table 2: NIAP Technical Decisions for CPP_FDE_AA_V2.0E

TD	Description	Applicable?	Non-applicability Rationale
TD1046	FIT Technical Decision: Addition of AES-XTS in FCS_COP.1(g)	Yes	
TD1042	FIT Technical Decision: Replacing FIPS 186-4 with 186-5	Yes	
TD0929	FIT Technical Decision: Clarification to FCS_PCC_EXT.1.1	Yes	
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	Yes	
TD0767	FIT Technical Decision for FMT_SMF.1.1	Yes	
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	Yes	
TD0765	IT Technical Decision for FMT_MOF.1	Yes	
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	Yes	
TD0759	IT Technical Decision for FCS_AFA_EXT.1.1	Yes	
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	Yes	
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	

The following table contains the TDs for the collaborative Protection Profile for Full Drive Encryption - Encryption Engine, Version 2.0E (CPP_FDE_EE_V2.0E) at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP Technical Decisions for CPP_FDE_EE_V2.0E

TD	Description	Applicable?	Non-applicability Rationale
TD1046	FIT Technical Decision: Addition of AES-XTS in FCS_COP.1(g)	No	The ST does not claim FCS_COP.1(g) in CPP_FDE_EE_V2.0E.
TD1042	FIT Technical Decision: Replacing FIPS 186-4 with 186-5	Yes	
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	Yes	
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	Yes	
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	Yes	
TD0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Yes	
TD0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Yes	
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	

3 Security Problem Definition

The threats, assumptions, and organizational security policies (OSPs) are defined in the documents specified in Section 2 "CC Conformance Claim". This Security Target includes by reference the Security Problem Definition (composed of threats, assumptions, and organizational security policies) from CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E.

4 Security Objectives

The Security Objectives for the TOE and Security Objectives for the Operational Environment are defined in the documents specified in Section 2 "CC Conformance Claim". This Security Target includes by reference the Security Objectives from CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E.

5 Extended Components Definition

All extended SFR components are defined in the PPs claimed in Section 2.

6 Security Requirements

6.1 Security Functional Requirements

The following conventions are used to indicate the operations performed within the ST on security requirement components:

- Selections are shown in **bold text** and are surrounded by square brackets.
- Assignments are shown in *italic text* and are surrounded by square brackets. The assignments within a selection are shown in ***bold italics***.
- Iterations are identified by appending a suffix to the original SFR.
- Refinements added to the text are shown in underlined text, deletions are shown as ~~strikethrough text~~.

6.1.1 Cryptographic Support (FCS)

6.1.1.1 FCS_AFA_EXT.1 – Authorization Factor Acquisition

Origin: CPP_FDE_AA_V2.0E

FCS_AFA_EXT.1.1 The TSF shall accept the following authorization factors [

- **a submask derived from a password authorization factor conditioned as defined in FCS_PCC_EXT.1**.

]

TSS Link: Section 7.1.1.1

6.1.1.2 FCS_AFA_EXT.2 – Timing of Authorization Factor Acquisition

Origin: CPP_FDE_AA_V2.0E

FCS_AFA_EXT.2.1 The TSF shall reacquire the authorization factor(s) specified in FCS_AFA_EXT.1 upon transition from any Compliant power saving state specified in FPT_PWR_EXT.1 prior to permitting access to plaintext data.

TSS Link: Section 7.1.1.2

6.1.1.3 FCS_CKM.1(b) – Cryptographic Key Generation (Symmetric Key)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_CKM.1.1(b) The TSF shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes [**256 bit**] that meet the following: [no standard].

TSS Link: Section 7.1.1.3

6.1.1.4 FCS_CKM.1(c) – Cryptographic Key Generation (Data Encryption Key)

Origin: CPP_FDE_EE_V2.0E

FCS_CKM.1.1(c) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation method [

- **generate a DEK using the RBG as specified in FCS_RBG_EXT.1**

] and specified cryptographic key sizes [**256 bits**].

TSS Link: Section 7.1.1.3

6.1.1.5 FCS_CKM.4(a)/AA – Cryptographic Key Destruction (Power Management)

Origin: CPP_FDE_AA_V2.0E

FCS_CKM.4.1(a)/AA The TSF shall [**erase**] cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1 that meets the following: [a key destruction method specified in FCS_CKM.4(d)].

TSS Link: Section 7.1.1.4

6.1.1.6 FCS_CKM.4(a)/EE – Cryptographic Key Destruction (Power Management)

Origin: CPP_FDE_EE_V2.0E

FCS_CKM.4.1(a)/EE The TSF shall [**erase**] cryptographic keys and key material from volatile memory when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1 that meets the following: [a key destruction method specified in FCS_CKM_EXT.6].

TSS Link: Section 7.1.1.4

6.1.1.7 FCS_CKM.4(b) – Cryptographic Key Destruction (TOE-Controlled Hardware)

Origin: CPP_FDE_EE_V2.0E

FCS_CKM.4.1(b) The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- **For volatile memory, the destruction shall be executed by a**
  - **single overwrite consisting of**
    - **zeroes**
  - **removal of power to the memory,**

] that meets the following: [no standard].

TSS Link: Section 7.1.1.4

6.1.1.8 FCS_CKM.4(d) – Cryptographic Key Destruction (Software TOE, 3rd Party Storage)

Origin: CPP_FDE_AA_V2.0E

FCS_CKM.4.1(d) The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- **For volatile memory, the destruction shall be executed by a**
  - **single overwrite consisting of**
    - **zeroes**
  - **removal of power to the memory,**

] that meets the following: [no standard].

TSS Link: Section 7.1.1.4

6.1.1.9 FCS_CKM_EXT.4(a) – Cryptographic Key and Key Material Destruction (Destruction Timing)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_CKM_EXT.4.1(a) The TSF shall destroy all keys and key material when no longer needed.

TSS Link: Section 7.1.1.4

6.1.1.10 FCS_CKM_EXT.4(b) – Cryptographic Key and Key Material Destruction (Power Management)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_CKM_EXT.4.1(b) The TSF shall destroy all key material, BEV, and authentication factors stored in plaintext when transitioning to a Compliant power saving state as defined by FPT_PWR_EXT.1.

TSS Link: Section 7.1.1.4

6.1.1.11 FCS_CKM_EXT.6 – Cryptographic Key Destruction Types

Origin: CPP_FDE_EE_V2.0E

FCS_CKM_EXT.6.1 The TSF shall use [**FCS_CKM.4(b)**] key destruction methods.

TSS Link: Section 7.1.1.4

6.1.1.12 FCS_COP.1(a) – Cryptographic Operation (Signature Verification)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

Applied TDs: TD1042

FCS_COP.1.1(a) The TSF shall perform [cryptographic signature services (verification)] in accordance with a [

- **RSA Digital Signature Algorithm with a key size (modulus) of 4096-bit;**

] that meet the following [

- **FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5, using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA schemes**

]

TSS Link: Section 7.1.1.5

6.1.1.13 FCS_COP.1(b) – Cryptographic Operation (Hash Algorithm)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_COP.1.1(b) The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [**SHA-256, SHA-384**] that meet the following: [ISO/IEC 10118-3:2004].

TSS Link: Section 7.1.1.6

6.1.1.14 FCS_COP.1(c)/AA – Cryptographic Operation (Keyed Hash Algorithm)

Origin: CPP_FDE_AA_V2.0E

FCS_COP.1.1(c)/AA The TSF shall perform cryptographic [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256**] and cryptographic key sizes [*256 bits*] that meet the following: [ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"].

TSS Link: Section 7.1.1.7

6.1.1.15 FCS_COP.1(c)/EE – Cryptographic Operation (Message Authentication)

Origin: CPP_FDE_EE_V2.0E

FCS_COP.1.1(c)/EE The TSF shall perform cryptographic [message authentication] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256**] and cryptographic key sizes [*256 bits used in* [**HMAC**]] that meet the following: [**ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"**].

TSS Link: Section 7.1.1.7

6.1.1.16 FCS_COP.1(d) – Cryptographic Operation (Key Wrapping)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_COP.1.1(d) The TSF shall perform [key wrapping] in accordance with a specified cryptographic algorithm [AES] in the following modes [**KW**] and the

cryptographic key size [**256 bits**] that meet the [AES as specified in ISO/IEC 18033-3, [**NIST SP 800-38F**]].

TSS Link: Section 7.1.1.8

6.1.1.17 FCS_COP.1(f) – Cryptographic Operation (AES Data Encryption/Decryption)

Origin: CPP_FDE_EE_V2.0E

FCS_COP.1.1(f) The TSF shall perform [data encryption and decryption] in accordance with a specified cryptographic algorithm [AES used in [**XTS**] mode] and cryptographic key sizes [**256 bits**] that meet the following: [AES as specified in ISO /IEC 18033-3**,** [**XTS as specified in IEEE 1619**]].

TSS Link: Section 7.1.1.9

6.1.1.18 FCS_COP.1(g) – Cryptographic Operation (Key Encryption)

Origin: CPP_FDE_AA_V2.0E

Applied TDs: TD1046

FCS_COP.1.1(g) The TSF shall perform [key encryption and decryption] in accordance with a specified cryptographic algorithm [AES used in [**CBC**] mode] and cryptographic key sizes [**256 bits**] that meet the following: [AES as specified in ISO /IEC 18033-3, [**CBC as specified in ISO/IEC 10116**]].

TSS Link: Section 7.1.1.10

6.1.1.19 FCS_KYC_EXT.1 – Key Chaining (Initiator)

Origin: CPP_FDE_AA_V2.0E

FCS_KYC_EXT.1.1 The TSF shall maintain a key chain of: [

- **one, using a submask as the BEV**

] while maintaining an effective strength of [**256 bits**] for symmetric keys and an effective strength of [**not applicable**] for asymmetric keys.

FCS_KYC_EXT.1.2 The TSF shall provide at least a [**256 bit**] BEV to [*the EE*] [

- **after the TSF has successfully performed the validation process as specified in FCS_VAL_EXT.1/AA~~FCS_VAL_EXT.1~~**

]

TSS Link: Section 7.1.1.11

6.1.1.20 FCS_KYC_EXT.2 – Key Chaining (Recipient)

Origin: CPP_FDE_EE_V2.0E

FCS_KYC_EXT.2.1 The TSF shall accept a BEV of at least [**256 bits**] from [the AA].

FCS_KYC_EXT.2.2 The TSF shall maintain a chain of intermediary keys originating from the BEV to the DEK using the following method(s): [

- **symmetric key generation as specified in FCS_CKM.1(b)**
- **key wrapping as specified in FCS_COP.1(d)**

] while maintaining an effective strength of [**256 bits**] for symmetric keys and an effective strength of [**not applicable**] for asymmetric keys.

TSS Link: Section 7.1.1.11

6.1.1.21 FCS_PCC_EXT.1 – Cryptographic Password Construct and Conditioning

Origin: CPP_FDE_AA_V2.0E

Applied TDs: TD0929

FCS_PCC_EXT.1.1 A password used by the TSF to generate a password authorization factor shall enable at least [*255*] characters in the set of {upper case characters, lower case characters, numbers, and [*all other 8-bit special characters*]} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[**SHA-256**], with [***an iteration count of 1***] **and** [***at least 50,000***] **subsequent rounds of AES operations with a device key and PBKDF2 output per FCS_COP.1(g) or FCS_COP.1(e)**], and output cryptographic key sizes [**256 bits**] that meet the following: [NIST SP 800-132].

TSS Link: Section 7.1.1.1

6.1.1.22 FCS_RBG_EXT.1 – Cryptographic Operation (Random Bit Generation)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with [[**NIST SP 800-90A**]] using [**CTR_DRBG (AES)**].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [

- [***1***] **hardware-based noise source(s)**

]

with a minimum of [**256 bits**] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

TSS Link: Section 7.1.1.12

6.1.1.23 FCS_SNI_EXT.1 – Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

Applied TDs: TD0760

FCS_SNI_EXT.1.1 The TSF shall [**use salts that are generated by** [**DRBG as specified in FCS_RBG_EXT.1**]].

FCS_SNI_EXT.1.2 The TSF shall use [**no nonces**].

FCS_SNI_EXT.1.3 The TSF shall [**create IVs in the following manner**

- **CBC: IVs shall be non-repeating and unpredictable;**
- **XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer;**

]

TSS Link: Section 7.1.1.13

6.1.1.24 FCS_VAL_EXT.1/AA – Validation

Origin: CPP_FDE_AA_V2.0E

FCS_VAL_EXT.1.1/AA The TSF shall perform validation of the [**BEV**] using the following method(s): [

- **key wrap as specified in FCS_COP.1(d)**

]

FCS_VAL_EXT.1.2/AA The TSF shall require validation of the [BEV] prior to [forwarding the BEV to the EE].

FCS_VAL_EXT.1.3/AA The TSF shall [

- **require power cycle/reset the TOE after** [***10***] **of consecutive failed validation attempts.**

]

TSS Link: Section 7.1.1.14

6.1.1.25 FCS_VAL_EXT.1/EE – Validation

Origin: CPP_FDE_EE_V2.0E

FCS_VAL_EXT.1.1/EE The TSF shall perform validation of the [BEV] using the following method(s):

[

- **key wrap as specified in FCS_COP.1(d)**

]

FCS_VAL_EXT.1.2/EE The TSF shall require validation of the [BEV] prior to [allowing access to TSF data after exiting a Compliant power saving state].

FCS_VAL_EXT.1.3/EE The TSF shall [

- **require power cycle/reset the TOE after** [***10***] **of consecutive failed validation attempts.**

]

TSS Link: Section 7.1.1.14

6.1.2 User Data Protection (FDP)

6.1.2.1 FDP_DSK_EXT.1 – Protection of Data on Disk

Origin: CPP_FDE_EE_V2.0E

FDP_DSK_EXT.1.1 The TSF shall perform Full Drive Encryption in accordance with FCS_COP.1(f), such that the drive contains no plaintext protected data.

FDP_DSK_EXT.1.2 The TSF shall encrypt all protected data without user intervention.

TSS Link: Section 7.1.2.1

6.1.3 Security Management (FMT)

6.1.3.1 FMT_MOF.1 – Management of Security Functions Behavior

Origin: CPP_FDE_AA_V2.0E

FMT_MOF.1.1 The TSF shall restrict the ability to [modify the behaviour of] the functions [use of Compliant power saving state] to [authorized users].

TSS Link: Section 7.1.3

6.1.3.2 FMT_SMF.1/AA – Specification of Management of Functions

Origin: CPP_FDE_AA_V2.0E

Applied TDs: TD0767

FMT_SMF.1.1/AA The TSF shall be capable of performing the following management functions: [

a) forward request to change the DEK to the EE

b) forwarding requests to cryptographically erase the DEK to the EE

c) allowing authorized users to change authorization values or set of authorization values used within the supported authorization method

d) initiate TOE firmware/software updates

e) [**no other functions**]

].

TSS Link: Section 7.1.3

6.1.3.3 FMT_SMF.1/EE – Specification of Management of Functions

Origin: CPP_FDE_EE_V2.0E

FMT_SMF.1.1/EE The TSF shall be capable of performing the following management functions: [

a) change the DEK, as specified in FCS_CKM.1, when re-provisioning or when commanded

b) erase the DEK, as specified in FCS_CKM.4(a)/EE ~~FCS_CKM.4(a)~~

c) initiate TOE firmware/software updates

d) [**no other functions**]

].

TSS Link: Section 7.1.3

6.1.3.4 FMT_SMR.1 – Security Roles

Origin: CPP_FDE_AA_V2.0E

FMT_SMR.1.1 The TSF shall maintain the roles [authorized user].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

TSS Link: Section 7.1.3

6.1.4 Protection of the TSF (FPT)

6.1.4.1 FPT_FUA_EXT.1 – Firmware Update Authentication

Origin: CPP_FDE_EE_V2.0E

FPT_FUA_EXT.1.1 The TSF shall authenticate the source of the firmware update using the digital signature algorithm specified in FCS_COP.1(a) using the RTU that contains the [**public key**].

FPT_FUA_EXT.1.2 The TSF shall only allow installation of update if the digital signature has been successfully verified as specified in FCS_COP.1(a).

FPT_FUA_EXT.1.3 The TSF shall only allow modification of the existing firmware after the successful validation of the digital signature, using a mechanism as described in FPT_TUD_EXT.1.2/EE ~~FPT_TUD_EXT.1.2~~.

FPT_FUA_EXT.1.4 The TSF shall return an error code if any part of the firmware update process fails.

TSS Link: Section 7.1.4.2

6.1.4.2 FPT_KYP_EXT.1 – Protection of Key and Key Material

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

TD applied: TD0458

FPT_KYP_EXT.1.1 The TSF shall [

- **only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1(d), or encrypted, as specified in FCS_COP.1(g) or FCS_COP.1(e)**
- **only store plaintext keys that meet any one of the following criteria**
  - **the plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1**

].

TSS Link: Section 7.1.4.1

6.1.4.3 FPT_PWR_EXT.1 – Power Saving States

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

Applied TDs: TD0464

FPT_PWR_EXT.1.1 The TSF shall define the following Compliant power saving states: [**G2(S5)**].

TSS Link: Section 7.1.4.3

6.1.4.4 FPT_PWR_EXT.2 – Timing of Power Saving States

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FPT_PWR_EXT.2.1 For each Compliant power saving state defined in FPT_PWR_EXT.1.1, the TSF shall enter the Compliant power saving state when the following conditions occur: user-initiated request, [**shutdown**].

TSS Link: Section 7.1.4.3

6.1.4.5 FPT_TST_EXT.1 – TSF Testing

Origin: CPP_FDE_AA_V2.0E, CPP_FDE_EE_V2.0E

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [**during initial start-up (on power on)**] to demonstrate the correct operation of the TSF: [

a) authenticity and integrity check of software/firmware

b) Known Answer Tests (KATs)

i. AES-CBC encrypt

ii. RSA signature verification

iii. HMAC-SHA-256 MAC generation

iv. PBKDF2 key derivation

v. CTR_DRBG health test

].

TSS Link: Section 7.1.4.4

6.1.4.6 FPT_TUD_EXT.1/AA – Trusted Update

Origin: CPP_FDE_AA_V2.0E

FPT_TUD_EXT.1.1/AA The TSF shall provide [authorized users] the ability to query the current version of the TOE [**software, firmware**].

FPT_TUD_EXT.1.2/AA The TSF shall provide [authorized users] the ability to initiate updates to TOE [**software, firmware**].

FPT_TUD_EXT.1.3/AA The TSF shall verify updates to the TOE software using a [digital signature as specified in FCS_COP.1(a)] by the manufacturer prior to installing those updates.

TSS Link: Section 7.1.4.2

6.1.4.7 FPT_TUD_EXT.1/EE – Trusted Update

Origin: CPP_FDE_EE_V2.0E

FPT_TUD_EXT.1.1/EE The TSF shall provide [authorized users] the ability to query the current version of the TOE [**software, firmware**].

FPT_TUD_EXT.1.2/EE The TSF shall provide [authorized users] the ability to initiate updates to TOE [**software, firmware**].

FPT_TUD_EXT.1.3/EE The TSF shall verify updates to the TOE [**software, firmware**] using an ~~a~~ [**authenticated firmware update mechanism as described in FPT_FUA_EXT.1**] by the manufacturer prior to installing those updates.

TSS Link: Section 7.1.4.2

6.2 Security Assurance Requirements

The Security Assurance Requirements (SARs) are included by reference from CPP_FDE_AA_V2.0E and CPP_FDE_EE_V2.0E. The table below summarizes the SARs for TOE evaluation.

Table 4: Assurance Requirements

Requirement Class	Requirement Components
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life-cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)

	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Tests (ATE)	Independent testing – Sample (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

6.2.1 ASE Security Target Evaluation

6.2.1.1 ASE_TSS.1 TOE summary specification

Developer action elements:

ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

Content and presentation elements:

ASE_TSS.1.1C

The TOE summary specification shall describe how the TOE meets each SFR, including a proprietary Key Management Description (Appendix E), and [**Entropy Essay**].

Evaluator action elements:

ASE_TSS.1.1E

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E

The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

7 TOE Summary Specification

7.1 TOE Security Functionality

As per the conformance claims defined in Section 2, the TOE supports the following security features:

- Cryptographic support
- User data protection
- Security management
- Protection of the TSF

7.1.1 Cryptographic Support (FCS)

The TOE uses the following cryptographic implementations:

- USR: Apple corecrypto Module 26 [Apple silicon, User, Software, SL1]
- KRN: Apple corecrypto Module 26 [Apple silicon, Kernel, Software, SL1]
- SKS: Apple corecrypto Module 26 [Apple silicon, Secure Key Store, Hardware, SL2]
- SE: Secure Enclave hardware onboard Apple silicon
- DMA: Storage Controller AES-XTS Engine hardware onboard Apple silicon

The USR, KRN, and SKS cryptographic implementations execute on the following operational environments:

- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M2
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M2 Pro
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M2 Max
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M2 Ultra
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M3
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M3 Pro
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M3 Max
- macOS Tahoe 26 on Apple M Series (ARMv8.6-A) M3 Ultra
- macOS Tahoe 26 on Apple M Series (ARMv9.2-A) M4
- macOS Tahoe 26 on Apple M Series (ARMv9.2-A) M4 Pro
- macOS Tahoe 26 on Apple M Series (ARMv9.2-A) M4 Max
- macOS Tahoe 26 on Apple M Series (ARMv9.3-A) M5

The SE cryptographic implementation executes on the following operational environments:

- Apple M Series (ARMv8.6-A) M2
- Apple M Series (ARMv8.6-A) M2 Pro
- Apple M Series (ARMv8.6-A) M2 Max
- Apple M Series (ARMv8.6-A) M2 Ultra
- Apple M Series (ARMv8.6-A) M3
- Apple M Series (ARMv8.6-A) M3 Pro
- Apple M Series (ARMv8.6-A) M3 Max

- Apple M Series (ARMv8.6-A) M3 Ultra
- Apple M Series (ARMv9.2-A) M4
- Apple M Series (ARMv9.2-A) M4 Pro
- Apple M Series (ARMv9.2-A) M4 Max
- Apple M Series (ARMv9.3-A) M5

The tables below show the cryptographic services used by the TOE and provided by the cryptographic implementations, describing the algorithms, their supported key sizes, applicable standard and purpose. The tables also include the certificates obtained from the Cryptographic Algorithm Validation Program (CAVP) in the evaluated configuration for each of the cryptographic algorithms.

Table 5: Mapping of SFRs to CAVP certificates (USR cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(a)	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384 Padding: PKCS#1 v1.5	[FIPS186-5]	A7643
FCS_COP.1(b)	SHA2-384	Byte-oriented mode	[FIPS180-4]	A7645

Table 6: Mapping of SFRs to CAVP certificates (KRN cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(a)	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384 Padding: PKCS#1 v1.5	[FIPS186-5]	A7635
FCS_COP.1(b)	SHA2-384	Byte-oriented mode	[FIPS180-4]	A7637

Table 7: Mapping of SFRs to CAVP certificates (SKS cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(a)	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384 Padding: PKCS#1 v1.5	[FIPS186-5]	A7652
FCS_COP.1(b)	SHA2-256, SHA2-384	Byte-oriented mode	[FIPS180-4]	A7652
FCS_COP.1(c)/AA	HMAC-SHA2-256	Byte-oriented mode	[FIPS198-1]	A7653
FCS_COP.1(c)/EE	HMAC-SHA2-256	Byte-oriented mode	[FIPS198-1]	A7653
FCS_COP.1(d)	AES-KW	256 bits encrypt, decrypt	[SP800-38F]	A7647, A7650
FCS_PCC_EXT.1	PBKDF2	HMAC-SHA2-256	[SP800-132]	A7649

Table 8: Mapping of SFRs to CAVP certificates (SE cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(g)	AES-CBC	256 bits encrypt	[SP800-38A]	A3496, A6547

FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A3490, A6548, A6924

Table 9: Mapping of SFRs to CAVP certificates (DMA cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1(f)	AES-XTS	256 bits encrypt, decrypt	[SP800-38E]	CCTL Tested

7.1.1.1 FCS_AFA_EXT.1 – Authorization Factor Acquisition, FCS_PCC_EXT.1 – Cryptographic Password Construction and Conditioning

The TOE accepts the password authorization factor from the user. Passwords of up to and including 255 characters are supported and can be comprised of any combination of uppercase characters, lowercase characters, numbers, and any other 8-bit special character.

To condition a user password, the Secure Enclave implements PBKDF2 as specified in NIST SP 800-132, following "Option 2b" defined in section 5.4 of the standard. It uses the HMAC-SHA-256 function as the pseudorandom function (PRF).

The inputs to PBKDF2 are:

- The 128-bit per-user salt generated by the TRNG.
- The user password without any pre-processing.
- An iteration count of one.

The output is a 256-bit intermediate value.

Next, this intermediate value is encrypted once using the System Measurement Device Key (SMDK) to bind it. Finally, the result is repeatedly encrypted with the AES-CBC-256 hardware cipher using the 256-bit Secure Enclave Unique ID (UID) as the encryption key to generate 256 bits of data with each loop iteration. The output, after all AES iterations have completed, forms a 256-bit key, which is termed Password-Derived Key (PDK). The resulting PDK is defined as the password submask and BEV in the TOE.

7.1.1.2 FCS_AFA_EXT.2 – Timing of Authorization Factor Acquisition

In order to exit the TOE from a Compliant power saving state, the user must reauthenticate with their username and password.

7.1.1.3 FCS_CKM.1(b) – Cryptographic Operation (Symmetric Keys), FCS_CKM.1(c) – Cryptographic Operation (Data Encryption Key)

At the time of creating a data volume, the TOE uses the random bit generator specified in FCS_RBG_EXT.1 to generate the following 256-bit AES keys:

- A Volume Encryption Key (VEK) for encrypting the volume, which serves as the Data Encryption Key (DEK).
- A Key Encryption Key (KEK) for protecting the VEK.

Volume and metadata contents are encrypted with the VEK, which is wrapped with the KEK.

7.1.1.4 FCS_CKM.4(a) – Cryptographic Key Destruction (Power Management), FCS_CKM.4(b) – Cryptographic Key Destruction (TOE-Controlled Hardware), FCS_CKM.4(d) – Cryptographic Key Destruction (Software TOE, 3rd Party Storage), FCS_CKM_EXT.4(a) – Cryptographic Key and Key Material Destruction (Destruction Timing), FCS_CKM_EXT.4(b) – Cryptographic Key and Key Material

Destruction (Destruction Timing), FCS_CKM_EXT.6 – Cryptographic Key Destruction Types

The Secure Enclave UID is fused into the Secure Enclave. The UID is not accessible by any component outside of the Secure Enclave and cannot be erased.

The following keys and key materials are stored in volatile memory (DRAM):

- Media Key
- System Measurement Root Key
- System Measurement Device Key
- Password-Derived Key (a.k.a. BEV)
- Root Encryption Key
- Key Encryption Key
- Volume Encryption Key
- DMA Storage Controller Key

Keys are introduced into volatile memory only when they are required to perform a specific cryptographic operation, by unwrapping the wrapped form stored in non-volatile memory. Since the keys are being used by the Secure Enclave to perform the operation, the Secure Enclave tracks the memory location of the key until the operation is complete. Once a key is no longer needed for the specific operation, the key is erased from volatile memory. The TOE erases cryptographic keys and key material from volatile memory by performing a single overwrite of zeroes and/or by removal of power to the memory.

The TOE will destroy all key material and cryptographic keys stored in plaintext when transitioning to a Compliant power saving state.

7.1.1.5 FCS_COP.1(a) – Cryptographic Operation (Signature Verification)

Signature verification is performed as part of the following features:

- Installing firmware/software updates
- Secure boot

Installation and secure boot signature verification involves different TOE components in different layers of the TOE and, thus, use the user space, kernel space, and SKS corecrypto implementations listed in Section 7.1.1.

The Boot ROM code contains the Apple Root CA public key, which is used to verify the digital signatures of the bootchain. The TOE cryptographically verifies each piece of software it loads in each step of boot procedure. Signatures are verified using RSA with a 4096-bit modulus, PKCS#1 v1.5 padding, and SHA2-384.

7.1.1.6 FCS_COP.1(b) – Cryptographic Operation (Hashing)

The TOE uses SHA2-256 in PBKDF2 to derive the PDK and uses SHA2-384 to verify digital signatures.

7.1.1.7 FCS_COP.1(c) – Cryptographic Operation (Message Authentication)

PBKDF2 uses HMAC-SHA-256 with a key length of 256 bits, a block size of 512 bits, and an output MAC length of 256 bits.

7.1.1.8 FCS_COP.1(d) - Cryptographic Operation (Key Wrapping)

The TOE performs key wrapping using AES in KW mode according to NIST SP 800-38F.

7.1.1.9 FCS_COP.1(f) – Cryptographic Operation (AES Data Encryption/Decryption)

The TOE performs data encryption / decryption using AES in XTS mode with a 256-bit key.

7.1.1.10 FCS_COP.1(g) – Cryptographic Operation (Key Encryption)

The TOE performs key encryption using AES in CBC mode with a 256-bit key.

7.1.1.11 FCS_KYC_EXT.1 – Key Chaining (Initiator), FCS_KYC_EXT.2 – Key Chaining (Recipient)

The TOE supports BEV sizes of 256 bits.

7.1.1.12 FCS_RBG_EXT.1 – Random Bit Generation

The TOE performs random bit generation as described in Section 6.1.1.22 and the proprietary Entropy Assessment Report.

7.1.1.13 FCS_SNI_EXT.1 – Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

The TOE generates salts and initialization vectors (IVs) using the DRBG.

For AES-XTS, the tweak value is the physical block number of the media on which the file is being written. The number is incremented based on the block number values.

7.1.1.14 FCS_VAL_EXT.1 – Validation

The TOE uses a hardware-enforced, persistent “counter lockbox” to validate the PDK. Counter lockboxes hold the entropy needed to unlock password-protected user data, as well as a password verifier value. During authentication, the PDK is used to derive a new password verifier value; if the derived password verifier value matches the stored password verifier value, authentication succeeds and the PDK is regarded as valid. Otherwise, the PDK is invalid. This indicator of the PDK validity also serves as the authentication result returned to macOS when authenticating a user.

The password can be used to exit a Compliant power saving state.

7.1.2 User Data Protection (FDP)

7.1.2.1 FDP_DSK_EXT.1 – Protection of Data on Disk

The TOE provides a dedicated AES-XTS crypto engine built into the DMA path between the flash storage and the main memory of the host platform. This DMA Storage Controller is placed in the middle of the data path between the application processor and the storage device.

The DMA Storage Controller performs the encryption/ decryption of the data prior to reaching the application processor or the storage. When a read operation is made, the data must first be decrypted by the DMA Storage Controller before the application processor has access to the data. When a write operation is made, the data is first encrypted by the DMA Storage Controller and then written to storage as a block of encrypted data. This arrangement ensures that standard methods of accessing the storage drive via the operating system will pass through these functions.

FDE support is always enabled. When FileVault is turned on, valid user credentials are also required during the boot process. Without valid credentials, the data volume of the TOE device remains encrypted and protected from unauthorized access, even if the physical storage device is removed and connected to another computer.

The entire user data volume is encrypted. The following parts of storage are not encrypted: partition table, Extensible Firmware Interface (EFI) service partition, Apple File System (APFS) container metadata,

recovery volumes, pre-boot volumes, virtual machine (VM) volumes, system volumes (protected by the signed system volume feature), and CoreDump partitions (if present).

7.1.3 Security Management (FMT)

The TOE supports only one Compliant power saving state: G2(S5): the system is powered down. This behavior cannot be modified by users.

The DEK can be erased and changed by erasing the appropriate volume in Disk Utility. This forces the TOE to cryptographically erase and generate a new DEK.

The password can be changed by navigating to System Settings > Users & Groups > Select the appropriate user > Change Password.

The updates to the TOE software/firmware are all bundled together as part of macOS updates. An administrator can check for and install updates by navigating to System Settings > General > Software Update.

7.1.4 Protection of the TSF (FPT)

7.1.4.1 FPT_KYP_EXT.1 – Protection of Key and Key Material

All symmetric keys stored in non-volatile memory are cryptographically wrapped using AES-KW.

7.1.4.2 FPT_FUA_EXT.1 – Firmware Update Authentication, FPT_TUD_EXT.1 – Trusted Update

TOE updates are contained within the macOS update package, provided by the Apple Update Server. The update package is cryptographically signed by Apple, using RSA with a 4096-bit modulus, PKCS#1 v1.5 padding, and SHA2-384. The Boot ROM code contains the Apple Root CA public key, which serves as the Root of Trust for Update (RTU).

7.1.4.3 FPT_PWR_EXT.1 – Power Saving States, FPT_PWR_EXT.2 – Timing of Power Saving States

The TOE supports only one Compliant power saving state: G2(S5): the system is powered down. This state is entered by shutting down the TOE device.

7.1.4.4 FPT_TST_EXT.1 – TSF Testing

Secure Boot

When the TOE device is turned on, the boot process starts with executing code from read-only memory referred to as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication and is implicitly trusted. Each following step of the boot process contains components that are cryptographically signed by Apple to enable integrity checking. The boot process proceeds only after verifying the integrity of the software at every step, which creates a chain of trust rooted in hardware.

Self-tests

Self-tests are performed by the cryptographic implementations included in the TOE. At start-up, the cryptographic implementations perform Known Answer Tests (KATs) to ensure the correctness of the cryptographic functions. These tests include the following:

- AES-CBC encrypt
- RSA signature verification
- HMAC-SHA-256 MAC generation
- PBKDF2 key derivation
- CTR_DRBG health test

A. Devices Covered by this Evaluation

The evaluated configuration includes the following Apple devices:

Table 10: Hardware Platforms

Marketing Name	Model Identifier	Processor (Micro Architecture)	Security Chip
MacBook Pro (14-inch, M5)	Mac17,2	M5 (ARMv9.3-A)	SEP v3.0
MacBook Air (15-inch, M4, 2025)	Mac16,13	M4 (ARMv9.2-A)	SEP v2.0
MacBook Air (13-inch, M4, 2025)	Mac16,12	M4 (ARMv9.2-A)	SEP v2.0
Mac Studio (2025)	Mac16,9	M4 Max (ARMv9.2-A)	SEP v2.0
Mac Studio (2025)	Mac15,14	M3 Ultra (ARMv8.6-A)	SEP v2.0
MacBook Pro (14-inch, 2024)	Mac16,8	M4 Pro (ARMv9.2-A)	SEP v2.0
	Mac16,6	M4 Max (ARMv9.2-A)	SEP v2.0
	Mac16,1	M4 (ARMv9.2-A)	SEP v2.0
MacBook Pro (16-inch, 2024)	Mac16,7	M4 Pro (ARMv9.2-A)	SEP v2.0
	Mac16,5	M4 Max (ARMv9.2-A)	SEP v2.0
iMac (24-inch, 2024, Two ports)	Mac16,3	M4 (ARMv9.2-A)	SEP v2.0
	Mac16,2	M4 (ARMv9.2-A)	SEP v2.0
Mac mini (2024)	Mac16,11	M4 Pro (ARMv9.2-A)	SEP v2.0
	Mac16,10	M4 (ARMv9.2-A)	SEP v2.0
MacBook Air (15-inch, M3, 2024)	Mac15,13	M3 (ARMv8.6-A)	SEP v2.0
MacBook Air (13-inch, M3, 2024)	Mac15,12	M3 (ARMv8.6-A)	SEP v2.0
MacBook Pro (14-inch, Nov 2023)	Mac15,10	M3 Max (ARMv8.6-A)	SEP v2.0
	Mac15,8	M3 Max (ARMv8.6-A)	SEP v2.0
	Mac15,6	M3 Pro (ARMv8.6-A)	SEP v2.0
	Mac15,3	M3 (ARMv8.6-A)	SEP v2.0
MacBook Pro (16-inch, Nov 2023)	Mac15,11	M3 Max (ARMv8.6-A)	SEP v2.0
	Mac15,9	M3 Max (ARMv8.6-A)	SEP v2.0
	Mac15,7	M3 Pro (ARMv8.6-A)	SEP v2.0
iMac (24-inch, 2023, Four ports)	Mac15,5	M3 (ARMv8.6-A)	SEP v2.0

iMac (24-inch, 2023, Two ports)	Mac15,4	M3 (ARMv8.6-A)	SEP v2.0
Mac Pro (-/Rack 2023)	Mac14,8	M2 Ultra (ARMv8.6-A)	SEP v2.0
MacBook Air (15-inch, 2023)	Mac14,15	M2 (ARMv8.6-A)	SEP v2.0
Mac Studio (2023)	Mac14,14	M2 Ultra (ARMv8.6-A)	SEP v2.0
	Mac14,13	M2 Max (ARMv8.6-A)	SEP v2.0
MacBook Pro (16-inch, 2023)	Mac14,6	M2 Max (ARMv8.6-A)	SEP v2.0
	Mac14,10	M2 Pro (ARMv8.6-A)	SEP v2.0
MacBook Pro (14-inch, 2023)	Mac14,5	M2 Max (ARMv8.6-A)	SEP v2.0
	Mac14,9	M2 Pro (ARMv8.6-A)	SEP v2.0
Mac mini (M2 Pro, 2023)	Mac14,12	M2 Pro (ARMv8.6-A)	SEP v2.0
Mac mini (M2, 2023)	Mac14,3	M2 (ARMv8.6-A)	SEP v2.0
MacBook Pro (13-inch, M2, 2022)	Mac14,7	M2 (ARMv8.6-A)	SEP v2.0
MacBook Air (M2, 2022)	Mac14,2	M2 (ARMv8.6-A)	SEP v2.0

B. Abbreviations and Terminology

AA

Authorization Acquisition

AES

Advanced Encryption Standard

APFS

Apple File System

BEV

Border Encryption Value

CBC

Cipher Block Chaining

DEK

Data Encryption Key

DMA

Direct Memory Access

EE

Encryption Engine

EFI

Extensible Firmware Interface

FDE

Full Drive Encryption

FIPS

Federal Information Processing Standards

IV

Initialization Vector

KEK

Key Encryption Key

KW

Key Wrap

LLB

Low-Level Bootloader

NIAP

National Information Assurance Partnership

NIST

National Institute of Standards and Technology

OSP

Organizational Security Policy

PBKDF

Password-Based Key Derivation Function

PDK

Password-Derived Key

PKCS

Public-Key Cryptography Standards

PRF

Pseudorandom Function

PSS

Probabilistic Signature Scheme

ROM

Read-only Memory

RSA

Rivest-Shamir-Adleman

SEP

Secure Enclave Processor

sepOS

SEP Operating System

SoC

System on Chip

SKS

Secure Key Store

SMDK

System Measurement Device Key

SMRK

System Measurement Root Key

SP

Special Publication

TD

Technical Decision

TOE

Target of Evaluation

TRNG

True Random Number Generator

UID

Unique ID

VEK

Volume Encryption Key

XTS

XEX (xor–encrypt–xor) Tweakable Block Ciphertext Stealing