Apple Inc.

Apple macOS 13 Ventura: FileVault Security Target

Version:	1.1
Status:	Released
Last Update:	2023-11-28
Validation Body:	NIAP
Validation ID:	VID11348
Classification:	Public

Trademarks

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

The following terms are trademarks or registered trademarks of Intel Corporation in the United States and/or other countries.

- Core™
- Intel®
- Xeon®

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

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

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

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2023-11-03	Alejandro Masino	First official version
1.1	2023-11-28	Alejandro Masino	Address comments from NIAP.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Apple macOS 13 Ventura: FileVault Security Target
Version:	1.1
Status:	Released
Date:	2023-11-28
Sponsor:	Apple Inc.
Developer:	Apple Inc.
Validation Body:	NIAP
Validation ID:	VID11348
Keywords:	Full drive encryption, encryption engine, authorization and acquisition

1.2 TOE Identification

The TOE is Apple macOS 13 Ventura: FileVault.

1.3 TOE Type

The TOE type is an authorization and encryption engine product.

1.4 TOE Overview

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document also describes the intended operational environment of the TOE and the functional and assurance requirements that the TOE meets.

The TOE is the Apple macOS 13 Ventura: FileVault full drive encryption product which supports an authorization acquisition and encryption engine. It is part of the macOS operating system. The macOS operating system is a Unix-based OS which leverages the Apple Secure Enclave, found in the Apple silicon System on a Chip (SoCs) and in the Apple T2 Security Chip, to perform full drive encryption. It also leverages an AES cryptographic implementation built in to the DMA Storage Controller chip. The operating system core is a POSIX-compliant operating system built on top of the XNU kernel with standard Unix facilities available from the command line interface.

The tested version of the TOE is:

- Apple macOS 13.2.1

1.5 TOE Description

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

1.5.1 Physical Boundary

The TOE includes both hardware and software running on the Macs listed in Appendix A.1 "Devices Covered by this Evaluation". These Macs are organized into the following two groups:

- Apple silicon Macs
- "Intel with T2" Macs

The Apple silicon Macs group represents all systems listed in Appendix A.1 that use an Apple silicon System on a Chip (SoC). The "Intel with T2" Macs group represents all systems listed in Appendix A.1 that use an Intel processor with the Apple T2 Security Chip. These groups have implementation differences as indicated in this document.

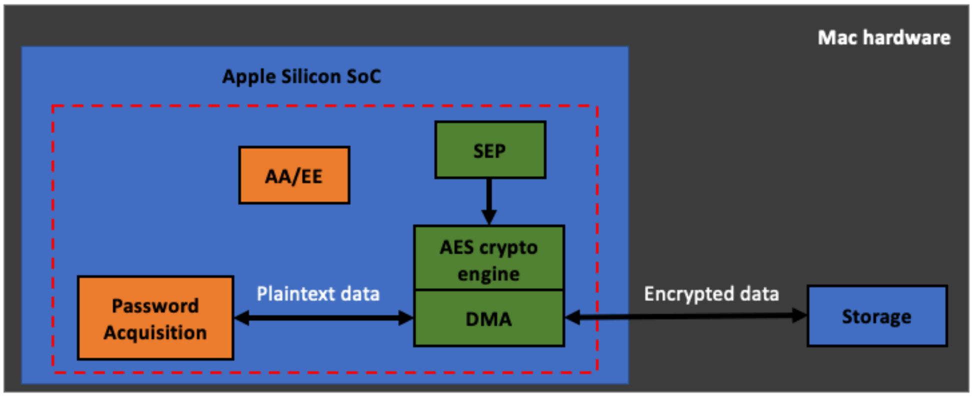
The TOE also includes the TOE documentation providing information for installing, configuring, and maintaining the evaluated configuration titled:

- Apple macOS 13 Ventura: FileVault Common Criteria Configuration Guide v1.0

1.5.1.1 Apple silicon

The Apple silicon SoC includes the application processor, which runs macOS, 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 Encryption Engine (EE) is instantiated in the Secure Enclave and the DMA Storage Controller. The AA is instantiated in both the application processor (Password Acquisition) of the SoC and the Secure Enclave. The Secure Enclave provides security related functionality for all the EE functionality (i.e., other than encryption/decryption of storage data) and all of the cryptographic functionality for AA (i.e., PBKDF2). The DMA Storage Controller provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between storage and main memory. The Password Acquisition component (AA) is the pre-boot component on the storage drive. It captures the user password and passes it to the Secure Enclave.

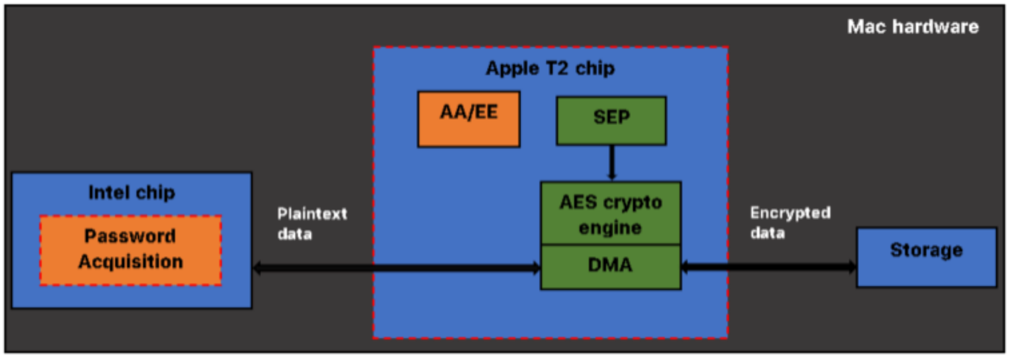
Figure 1: Apple silicon: Major components of TOE within red border



1.5.1.2 Intel with T2

The Apple T2 Security Chip runs the T2OS 13 operating system. The T2 includes the Secure Enclave, which contains the SEP running the sepOS operating system, and the DMA Storage Controller, which performs the storage encryption. The Encryption Engine (EE) is instantiated on the T2. The AA is instantiated on both the Intel chip (Password Acquisition) and the T2. The Secure Enclave provides the security related functionality for all the EE functionality (i.e., other than encryption/decryption of storage data) and all of the cryptographic functionality for AA (i.e., PBKDF2). The DMA Storage Controller provides a dedicated AES crypto engine built into the Direct Memory Access (DMA) path between the storage and main memory of the host platform. The Password Acquisition component (AA) is the pre-boot component on the storage drive. It captures the user password and passes it to the Secure Enclave.

Figure 2: Intel with T2: Major components of TOE within red border



1.5.1.3 Secure Enclave

The Secure Enclave is a dedicated secure subsystem integrated into the Apple silicon SoC and in the Apple T2 Security Chip. 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 engine. The TRNG and AES engine are directly connected to the SEP and are only accessible through the SEP.

Each SEP is provisioned during fabrication with its own 256-bit Unique ID (UID). This UID is:

- Used as a key by the device
- Not accessible to other parts of the system
- Not known to Apple

1.5.2 TOE Security Functionality

The following subsections describe the TOE's general security functionality.

1.5.2.1 Cryptographic Support (FCS)

1.5.2.1.1 Cryptographic modules

The TOE uses the following cryptographic modules to satisfy the cryptographic requirements defined in this ST:

- Apple silicon
    - Apple corecrypto Module 13.0 [Apple ARM, User, Software, SL1]
    - Apple corecrypto Module 13.0 [Apple ARM, Kernel, Software, SL1]
    - Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2]
    - Apple DMA Storage Controller 2.0 [Hardware]
- Intel with T2
    - Apple corecrypto Module 13.0 [Intel, User, Software, SL1]
    - Apple corecrypto Module 13.0 [Intel, Kernel, Software, SL1]
    - Apple corecrypto Module 13.0 [Apple ARM, Secure Key Store, Hardware, SL2]
    - Apple DMA Storage Controller 1.0 [Hardware]

On Apple silicon Macs, the Apple corecrypto Module 13.0 [Apple ARM, User, Software, SL1] module resides in the macOS user space. The Apple corecrypto Module 13.0 [Apple ARM, Kernel, Software, SL1] module resides in the macOS kernel space. The Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2] module resides in the Secure Enclave. The Apple DMA Storage Controller 2.0 [Hardware] module resides in the DMA Storage Controller.

On "Intel with T2" Macs, the Apple corecrypto Module 13.0 [Intel, User, Software, SL1] module resides in the macOS user space. The Apple corecrypto Module 13.0 [Intel, Kernel, Software, SL1] module resides in the macOS kernel space. The Apple corecrypto Module 13.0 [Apple ARM, Secure Key Store, Hardware, SL2] module resides in the Secure Enclave. The Apple DMA Storage Controller 1.0 [Hardware] module resides in the DMA Storage Controller.

Table 1 lists the cryptographic algorithms claimed in this evaluation along with their respective standards.

Table 1: CAVP References

Algorithms	Standards
AES	AES-CBC (as defined in NIST SP 800-38A)
AES	AES-KW (AES as specified in ISO/IEC 18033-3, [NIST SP 800-38F])
AES	AES-XTS (AES as specified in ISO/IEC18033-3 and XTS as specified in IEEE 1619)
ECDSA	FIPS PUB 186-4 Digital Signature Standard (DSS), Section 6 and Appendix D
RSA	FIPS PUB 186-4 Digital Signature Standard (DSS), Appendix B.3
HMAC	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"
SHS	NIST FIPS Pub 180-4
DRBG	CTR_DRBG (AES)

1.5.2.2 User Data Protection (FDP)

The TOE encrypts all user data using the following algorithms:

- Apple silicon: AES-XTS-256 using two independent 256-bit keys
- Intel with T2: AES-XTS-128 using two independent 128-bit keys

1.5.2.3 Security Management (FMT)

The TOE can perform management functions. The administrator has full access to carry out all management functions and the user have limited privilege. The System Settings » Privacy & Security menu on macOS invokes management functionality of the AA component.

1.5.2.4 Protection of the TSF (FPT)

The TOE implements the following protection of TSF data:

- Protection of Key and Key Material
- Power Saving States
- Timing of Power Saving States
- TSF Testing
- Trusted 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 signature for the package before it is installed.

1.5.3 TOE Operational Environment

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

Table 2: TOE operational environment

Component	Description
Hardware platform	See Table 8
Apple Update Server	Server that allows the TOE to download updates

1.5.4 Product Functionality Excluded from the Scope of the Evaluation

The following product functionality is not included in the CC evaluation.

- Biometric Authentication—Many Apple Macs support biometric authentication. This feature is outside the scope of the evaluation.

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 extended. Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

This Security Target claims exact conformance to the following Protection Profiles:

- [FDE_AA]☝: collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition, version 2.0 + Errata 20190201 as of 2019-02-01.
- [FDE_EE]☝: collaborative Protection Profile for Full Drive Encryption - Encryption Engine, version 2.0 + Errata 20190201, as of 2019-02-01.

The following sections describes the use cases that each document covers and the technical decisions applied.

2.1 collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition [FDE_AA]

Table 3 contains the NIAP Technical Decisions (TDs) for this protection profile at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP TDs for FDE_AAv2.0e

NIAP TD	TD description	Applicable?	Applicability rationale
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	No	The TOE does not claim any of the modified items.
TD0767	FIT Technical Decision for FMT_SMF.1.1	Yes	Modifies an SFR and evaluation activity wording.
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	No	The TOE does not claim any of the modified items.
TD0765	FIT Technical Decision for FMT_MOF.1	Yes	Modifies evaluation activities used by the evaluation.
TD0764	FIT Technical Decision for FCS_PCC_EXT.1	Yes	Modifies SFR wording used in this document.
TD0760	FIT Technical Decision for FCS_SNI_EXT.1.3, FCS_COP.1(f)	Yes	Modifies an SFR and evaluation activity wording.
TD0759	FIT Technical Decision for FCS_AFA_EXT.1.1	No	The TOE does not include the use of smartcards.
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No	The TOE is not a NAS device.
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	Modifies evaluation activities used by the evaluation.

2.2 collaborative Protection Profile for Full Drive Encryption - Encryption Engine [FDE_EE]

Table 4 contains the NIAP Technical Decisions (TDs) for this protection profile at the time of the evaluation and a statement of applicability to the evaluation.

Table 4: NIAP TDs for FDE_EEv2.0e

NIAP TD	TD description	Applicable?	Applicability rationale
TD0769	FIT Technical Decision for FPT_KYP_EXT.1.1	No	The TOE does not claim any of the modified items.
TD0766	FIT Technical Decision for FCS_CKM.4(d) Test Notes	No	The TOE does not claim any of the modified items.
TD0606	FIT Technical Recommendation for Evaluating a NAS against the FDE AA and FDEE	No	The TOE is not a NAS device.
TD0464	FIT Technical Decision for FPT_PWR_EXT.1 compliant power saving states	Yes	Modifies SFR wording used in this document.
TD0460	FIT Technical Decision for FPT_PWR_EXT.1 non-compliant power saving states	Yes	Modifies an evaluation activity used by the evaluation.
TD0458	FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities	Yes	Modifies evaluation activities used by the evaluation.

3 Security Problem Definition

The security problem definition has been taken from [FDE EE v2.0e] and [FDE AA v2.0e] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threat Environment

3.1.1 Threats countered by the TOE

T.UNAUTHORIZED_DATA_ACCESS

PP Origin: *FDE_AA, FDE_EE*

The cPP addresses the primary threat of unauthorized disclosure of protected data stored on a storage device. If an adversary obtains a lost or stolen storage device (e.g., a storage device contained in a laptop or a portable external storage device), they may attempt to connect a targeted storage device to a host of which they have complete control and have raw access to the storage device (e.g., to specified disk sectors, to specified blocks).

T.KEYING_MATERIAL_COMPROMISE/AA

PP Origin: *FDE_AA*

Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. The cPP considers possession of key material of equal importance to the data itself. Threat agents may look for key material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash.

T.KEYING_MATERIAL_COMPROMISE/EE

PP Origin: *FDE_EE*

Possession of any of the keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption. The cPP considers possession of keying material of equal importance to the data itself. Threat agents may look for keying material in unencrypted sectors of the storage device and on other peripherals in the operating environment (OE), e.g. BIOS configuration, SPI flash, or TPMs.

T.AUTHORIZATION_GUESSING/AA

PP Origin: *FDE_AA*

Threat agents may exercise host software to repeatedly guess authorization factors, such as passwords and PINs. Successful guessing of the authorization factors may cause the TOE to release BEV or otherwise put it in a state in which it discloses protected data to unauthorized users.

T.AUTHORIZATION_GUESSING/EE

PP Origin: *FDE_EE*

Threat agents may exercise host software to repeatedly guess authorization factors, such as passwords and PINs. Successful guessing of the authorization factors may cause the TOE to release DEKs or otherwise put it in a state in which it discloses protected data to unauthorized users.

T.KEYSPACE_EXHAUST

PP Origin: *FDE_AA, FDE_EE*

Threat agents may perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms and/or parameters allow attackers to exhaust the key space through brute force and give them unauthorized access to the data.

T.KNOWN_PLAINTEXT

PP Origin: *FDE_EE*

Threat agents know plaintext in regions of storage devices, especially in uninitialized regions (all zeroes) as well as regions that contain well known software such as operating systems. A poor choice of encryption algorithms, encryption modes, and initialization vectors along with known plaintext could allow an attacker to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.

T.CHOSEN_PLAINTEXT

PP Origin: *FDE_EE*

Threat agents may trick authorized users into storing chosen plaintext on the encrypted storage device in the form of an image, document, or some other file. A poor choice of encryption algorithms, encryption modes, and initialization vectors along with the chosen plaintext could allow attackers to recover the effective DEK, thus providing unauthorized access to the previously unknown plaintext on the storage device.

T.UNAUTHORIZED_UPDATE/AA

PP Origin: *FDE_AA*

Threat agents may attempt to perform an update of the product which compromises the security features of the TOE. Poorly chosen update protocols, signature generation and verification algorithms, and parameters may allow attackers to install software and/or firmware that bypasses the intended security features and provides them unauthorized access to data.

T.UNAUTHORIZED_UPDATE/EE

PP Origin: *FDE_EE*

Threat agents may attempt to perform an update of the product which compromises the security features of the TOE. Poorly chosen update protocols, signature generation and verification algorithms, and parameters may allow attackers to install software that bypasses the intended security features and provides them unauthorized access to data.

T.UNAUTHORIZED_FIRMWARE_UPDATE

PP Origin: *FDE_EE*

An attacker attempts to replace the firmware on the SED via a command from the AA or from the host platform with a malicious firmware update that may compromise the security features of the TOE.

T.UNAUTHORIZED_FIRMWARE_MODIFY

PP Origin: *FDE_EE*

An attacker attempts to modify the firmware in the SED via a command from the AA or from the host platform that may compromise the security features of the TOE.

3.2 Assumptions

A.INITIAL_DRIVE_STATE/AA

PP Origin: *FDE_AA*

Users enable Full Drive Encryption on a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption. The cPP does not intend to include requirements to find all the areas on storage devices that potentially contain protected data. In some cases, it may not be possible - for example, data contained in "bad" sectors.

While inadvertent exposure to data contained in bad sectors or un-partitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors, un-partitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE pre-authentication software) contain no protected data.

A.INITIAL_DRIVE_STATE/EE

PP Origin: *FDE_EE*

Users enable Full Drive Encryption on a newly provisioned storage device free of protected data in areas not targeted for encryption. It is also assumed that data intended for protection should not be on the targeted storage media until after provisioning. The cPP does not intend to include requirements to find all the areas on storage devices that potentially contain protected data. In some cases, it may not be possible - for example, data contained in "bad" sectors. While inadvertent exposure to data contained in bad sectors or un24 partitioned space is unlikely, one may use forensics tools to recover data from such areas of the storage device. Consequently, the cPP assumes bad sectors, un-partitioned space, and areas that must contain unencrypted code (e.g., MBR and AA/EE pre-authentication software) contain no protected data.

A.SECURE_STATE

PP Origin: *FDE_AA*

Upon the completion of proper provisioning, the drive is only assumed secure when in a powered off state up until it is powered on and receives initial authorization.

A.TRUSTED_CHANNEL

PP Origin: *FDE_AA, FDE_EE*

Communication among and between product components (e.g., AA and EE) is sufficiently protected to prevent information disclosure. In cases in which a single product fulfils both cPPs, then the communication between the components does not extend beyond the boundary of the TOE (e.g., communication path is within the TOE boundary). In cases in which independent products satisfy the requirements of the AA and EE, the physically close proximity of the two products during their operation means that the threat agent has very little opportunity to interpose itself in the channel between the two without the user noticing and taking appropriate actions.

A.TRAINED_USER/AA

PP Origin: *FDE_AA*

Authorized users follow all provided user guidance, including keeping password/passphrases and external tokens securely stored separately from the storage device and/or platform.

A.TRAINED_USER/EE

PP Origin: *FDE_EE*

Users follow the provided guidance for securing the TOE and authorization factors. This includes conformance with authorization factor strength, using external token authentication factors for no other purpose and ensuring external token authorization factors are securely stored separately from the storage device and/or platform. The user should also be trained on how to power off their system.

A.PLATFORM_STATE

PP Origin: *FDE_AA, FDE_EE*

The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.

A.SINGLE_USE_ET

PP Origin: *FDE_AA*

External tokens that contain authorization factors are used for no other purpose than to store the external token authorization factors.

A.POWER_DOWN/AA

PP Origin: *FDE_AA*

The user does not leave the platform and/or storage device unattended until all volatile memory is cleared after a power-off, so memory remnant attacks are infeasible.

Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., lock screen). Users power the platform and/or storage device down or place it into a power managed state, such as a "hibernation mode".

A.POWER_DOWN/EE

PP Origin: *FDE_EE*

The user does not leave the platform and/or storage device unattended until the device is in a Compliant power saving state or has fully powered off. This properly clears memories and locks down the device. Authorized users do not leave the platform and/or storage device in a mode where sensitive information persists in non-volatile storage (e.g., lock screen or sleep state). Users power the platform and/or storage device down or place it into a power managed state, such as a "hibernation mode".

A.PASSWORD_STRENGTH

PP Origin: *FDE_AA*

Authorized administrators ensure password/passphrase authorization factors have sufficient strength and entropy to reflect the sensitivity of the data being protected.

A.PLATFORM_I&A

PP Origin: *FDE_AA*

The product does not interfere with or change the normal platform identification and authentication functionality such as the operating system login. It may provide authorization factors to the operating system's login interface, but it will not change or degrade the functionality of the actual interface.

A.STRONG_CRYPTO

PP Origin: *FDE_AA, FDE_EE*

All cryptography implemented in the Operational Environment and used by the product meets the requirements listed in the cPP. This includes generation of external token authorization factors by a RBG.

A.PHYSICAL

PP Origin: *FDE_AA, FDE_EE*

The platform is assumed to be physically protected in its Operational Environment and not subject to physical attacks that compromise the security and/or interfere with the platform's correct operation.

4 Security Objectives

4.1 Objectives for the TOE

This ST does not define security objectives for the TOE.

4.2 Objectives for the Operational Environment

OE.TRUSTED_CHANNEL

PP Origin: *FDE_AA, FDE_EE*

Communication among and between product components (i.e., AA and EE) is sufficiently protected to prevent information disclosure.

OE.INITIAL_DRIVE_STATE

PP Origin: *FDE_AA, FDE_EE*

The OE provides a newly provisioned or initialized storage device free of protected data in areas not targeted for encryption.

OE.PASSPHRASE_STRENGTH

PP Origin: *FDE_AA, FDE_EE*

An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to guidance from the Enterprise using the TOE.

OE.POWER_DOWN/AA

PP Origin: *FDE_AA*

Volatile memory is cleared after power-off so memory remnant attacks are infeasible.

OE.POWER_DOWN/EE

PP Origin: *FDE_EE*

Volatile memory is cleared after entering a Compliant power saving state or turned off so memory remnant attacks are infeasible.

OE.SINGLE_USE_ET

PP Origin: *FDE_AA, FDE_EE*

External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.

OE.STRONG_ENVIRONMENT_CRYPTO

PP Origin: *FDE_AA, FDE_EE*

The Operating Environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE and Appendix A.

OE.TRAINED_USERS

PP Origin: *FDE_AA, FDE_EE*

Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.

OE.PLATFORM_STATE

PP Origin: *FDE_AA*

The platform in which the storage device resides (or an external storage device is connected) is free of malware that could interfere with the correct operation of the product.

OE.PLATFORM_I&A

PP Origin: *FDE_AA*

The Operational Environment will provide individual user identification and authentication mechanisms that operate independently of the authorization factors used by the TOE.

OE.PHYSICAL

PP Origin: *FDE_AA, FDE_EE*

The Operational Environment will provide a secure physical computing space such than an adversary is not able to make modifications to the environment or to the TOE itself.

4.3 Security Objectives Rationale

The rationale are defined in the documents specified in Section 2 "CC Conformance Claim".

5 Extended Components Definition

The extended components definitions are defined in the documents specified in Section 2 "CC Conformance Claim".

6 Security Requirements

6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~strikethrough~~ text.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Table 5: SFRs for the TOE

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FCS - Cryptographic support	FCS_AFA_EXT.1 Authorization Factor Acquisition		FDE_AA	No	No	No	Yes
	FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition		FDE_AA	No	Yes	No	No
	FCS_CKM.1(b) Cryptographic Key Generation (Symmetric Keys)	FCS_CKM.1	FDE_AA, FDE_EE	No	No	No	Yes
	FCS_CKM.1(c) Cryptographic Key Generation (Data Encryption Key)	FCS_CKM.1	FDE_EE	No	No	No	Yes
	FCS_CKM.4(a)/AA Cryptographic Key Destruction (Power Management) - Authorization Acquisition	FCS_CKM.4	FDE_AA	Yes	Yes	No	Yes
	FCS_CKM.4(a)/EE Cryptographic Key Destruction (Power Management) - Encryption Engine	FCS_CKM.4	FDE_EE	Yes	Yes	No	Yes
	FCS_CKM.4(b) Cryptographic Key Destruction (TOE-Controlled Hardware)	FCS_CKM.4	FDE_EE	No	No	No	Yes
	FCS_CKM.4(d) Cryptographic Key Destruction (Software TOE, 3rd Party Storage)	FCS_CKM.4	FDE_AA	No	No	No	Yes
	FCS_CKM_EXT.4(a) Cryptographic Key and Key Material Destruction (Destruction Timing)	FCS_CKM_EXT.4	FDE_AA, FDE_EE	No	No	No	No
	FCS_CKM_EXT.4(b) Cryptographic Key and Key Material Destruction (Power Management)	FCS_CKM_EXT.4	FDE_AA, FDE_EE	No	Yes	No	No
	FCS_CKM_EXT.6 Cryptographic Key Destruction Types		FDE_EE	No	No	No	Yes
	FCS_COP.1(a) Cryptographic Operation (Signature Verification)	FCS_COP.1	FDE_AA, FDE_EE	No	No	No	Yes

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FCS_COP.1(b) Cryptographic Operation (Hash Algorithm)	FCS_COP.1	FDE_AA, FDE_EE	No	No	No	Yes
	FCS_COP.1(c)/AA Cryptographic Operation (Keyed Hash Algorithm)	FCS_COP.1	FDE_AA	Yes	No	Yes	Yes
	FCS_COP.1(c)/EE Cryptographic Operation (Message Authentication)	FCS_COP.1	FDE_EE	Yes	No	Yes	Yes
	FCS_COP.1(d) Cryptographic Operation (Key Wrapping)	FCS_COP.1	FDE_AA, FDE_EE	No	No	No	Yes
	FCS_COP.1(f) Cryptographic Operation (AES Data Encryption/Decryption)	FCS_COP.1	FDE_EE	No	No	No	Yes
	FCS_COP.1(g) Cryptographic Operation (Key Encryption)	FCS_COP.1	FDE_AA	No	No	No	Yes
	FCS_KDF_EXT.1 Cryptographic Key Derivation		FDE_AA	No	Yes	No	Yes
	FCS_KYC_EXT.1 Key Chaining (Initiator)		FDE_AA	No	Yes	Yes	Yes
	FCS_KYC_EXT.2 Key Chaining (Recipient)		FDE_EE	No	No	No	Yes
	FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning		FDE_AA	No	No	Yes	Yes
	FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)		FDE_AA, FDE_EE	No	No	Yes	Yes
	FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)		FDE_AA, FDE_EE	No	No	No	Yes
	FCS_VAL_EXT.1/AA Validation	FCS_VAL_EXT.1	FDE_AA	Yes	Yes	Yes	Yes
	FCS_VAL_EXT.1/EE Validation	FCS_VAL_EXT.1	FDE_EE	Yes	Yes	Yes	Yes
FDP - User data protection	FDP_DSK_EXT.1 Protection of Data on Disk		FDE_EE	No	No	No	No
FMT - Security management	FMT_MOF.1 Management of Functions Behavior		FDE_AA	No	No	No	No
	FMT_SMF.1/AA Specification of Management Functions - Authorization Acquisition	FMT_SMF.1	FDE_AA	Yes	No	No	Yes
	FMT_SMF.1/EE Specification of Management Functions - Encryption Engine	FMT_SMF.1	FDE_EE	Yes	Yes	No	Yes
	FMT_SMR.1 Security Roles - Authorization Acquisition		FDE_AA	No	No	No	No

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FPT - Protection of the TSF	FPT_FUA_EXT.1 Firmware Update Authentication		FDE_EE	No	Yes	No	Yes
	FPT_KYP_EXT.1/AA Protection of Key and Key Material (AA)	FPT_KYP_EXT.1	FDE_AA	Yes	No	No	Yes
	FPT_KYP_EXT.1/EE Protection of Key and Key Material (EE)	FPT_KYP_EXT.1	FDE_EE	Yes	No	No	Yes
	FPT_PWR_EXT.1/AA Power Saving States (AA)	FPT_PWR_EXT.1	FDE_AA	Yes	No	No	Yes
	FPT_PWR_EXT.1/EE Power Saving States (EE)	FPT_PWR_EXT.1	FDE_EE	Yes	No	No	Yes
	FPT_PWR_EXT.2 Timing of Power Saving States		FDE_AA, FDE_EE	No	Yes	No	Yes
	FPT_TST_EXT.1 Testing		FDE_AA, FDE_EE	No	No	Yes	Yes
	FPT_TUD_EXT.1/AA Trusted Update	FPT_TUD_EXT.1	FDE_AA	Yes	No	No	Yes
	FPT_TUD_EXT.1/EE Trusted Update	FPT_TUD_EXT.1	FDE_EE	Yes	No	No	Yes

6.1.1 Cryptographic support (FCS)

6.1.1.1 FCS_AFA_EXT.1 Authorization Factor Acquisition

PP Origin: *FDE_AA*

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: *TSS for FCS_AFA_EXT.1*

6.1.1.2 FCS_AFA_EXT.2 Timing of Authorization Factor Acquisition

PP Origin: *FDE_AA*

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/AA ~~FPT_PWR_EXT.1~~ prior to permitting access to plaintext data.

TSS Link: *TSS for FCS_AFA_EXT.2*

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

PP Origin: *FDE_AA, FDE_EE*

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: *TSS for FCS_CKM.1(b)*

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

PP Origin: *FDE_EE*

FCS_CKM.1.1(c)

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm method

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

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

TSS Link: *TSS for FCS_CKM.1(c)*

6.1.1.5 FCS_CKM.4(a)/AA Cryptographic Key Destruction (Power Management) - Authorization Acquisition

PP Origin: *FDE_AA*

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/AA ~~FPT_PWR_EXT.1~~ that meets the following: [a key destruction method specified in FCS_CKM.4(d)].

TSS Link: *TSS for FCS_CKM.4(a)/AA*

6.1.1.6 FCS_CKM.4(a)/EE Cryptographic Key Destruction (Power Management) - Encryption Engine

PP Origin: *FDE_EE*

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/EE ~~FPT_PWR_EXT.1~~ that meets the following: [a key destruction method specified in FCS_CKM_EXT.6].

TSS Link: *TSS for FCS_CKM.4(a)/EE*

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

PP Origin: *FDE_EE*

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**
- **For non-volatile memory**
    - **that employs a wear-leveling algorithm, the destruction shall be executed by a**
        - **single overwrite consisting of zeroes**

that meets the following: [no standard].

TSS Link: *TSS for FCS_CKM.4(b)*

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

PP Origin: *FDE_AA*

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**

TSS Link: *TSS for FCS_CKM.4(d)*

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

PP Origin: *FDE_AA, FDE_EE*

FCS_CKM_EXT.4.1(a)

The TSF shall destroy all keys and key material when no longer needed.

TSS Link: *TSS for FCS_CKM_EXT.4(a)*

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

PP Origin: *FDE_AA, FDE_EE*

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/AA *and* FPT_PWR_EXT.1/EE ~~FPT_PWR_EXT.1~~.

TSS Link: *TSS for FCS_CKM_EXT.4(b)*

6.1.1.11 FCS_CKM_EXT.6 Cryptographic Key Destruction Types

PP Origin: *FDE_EE*

FCS_CKM_EXT.6.1

The TSF shall use FCS_CKM.4(b) key destruction methods.

TSS Link: *TSS for FCS_CKM_EXT.6*

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

PP Origin: *FDE_AA, FDE_EE*

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 ;**
- **Elliptic Curve Digital Signature Algorithm with a key size of 256 bits or greater**

that meet the following

- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 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**
- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-521 ; ISO/IEC 14888-3, Section 6.4, for ECDSA schemes**

TSS Link: *TSS for FCS_COP.1(a)*

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

PP Origin: *FDE_AA, FDE_EE*

FCS_COP.1.1(b)

The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm **SHA-256**, **SHA-512** that meet the following: [ISO/IEC 10118-3:2004].

TSS Link: *TSS for FCS_COP.1(b)*

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

PP Origin: *FDE_AA*

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 used in HMAC** that meet the following: [ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"].

TSS Link: *TSS for FCS_COP.1(c)/AA*

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

PP Origin: *FDE_EE*

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: *TSS for FCS_COP.1(c)/EE*

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

PP Origin: *FDE_AA, FDE_EE*

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 following: [AES as specified in ISO/IEC 18033-3, **NIST SP 800-38F**].

TSS Link: *TSS for FCS_COP.1(d)*

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

PP Origin: *FDE_EE*

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 **128 bits, 256 bits** that meet the following: [AES as specified in ISO /IEC 18033-3, **XTS as specified in IEEE 1619**].

TSS Link: *TSS for FCS_COP.1(f)*

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

PP Origin: *FDE_AA*

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: *TSS for FCS_COP.1(g)*

6.1.1.19 FCS_KDF_EXT.1 Cryptographic Key Derivation

PP Origin: *FDE_AA*

FCS_KDF_EXT.1.1

The TSF shall accept **a conditioned password submask** to derive an intermediate key, as defined in

- **NIST SP 800-132**

using the keyed-hash functions specified in FCS_COP.1(c)/AA ~~FCS_COP.1(c)~~, such that the output is at least of equivalent security strength (in number of bits) to the BEV.

TSS Link: *TSS for FCS_KDF_EXT.1*

6.1.1.20 FCS_KYC_EXT.1 Key Chaining (Initiator)

PP Origin: *FDE_AA*

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 **EE**

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

TSS Link: *TSS for FCS_KYC_EXT.1*

6.1.1.21 FCS_KYC_EXT.2 Key Chaining (Recipient)

PP Origin: *FDE_EE*

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: *TSS for FCS_KYC_EXT.2*

6.1.1.22 FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning

PP Origin: *FDE_AA*

Applied TDs: *TD0764*

FCS_PCC_EXT.1.1

A password used by the TSF to generate a password authorization factor shall enable up to **255** characters in the set of {upper case characters, lower case characters, numbers, and **all other 8-bit values**} 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: *TSS for FCS_PCC_EXT.1*

6.1.1.23 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

PP Origin: *FDE_AA, FDE_EE*

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

- **24 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: *TSS for FCS_RBG_EXT.1*

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

PP Origin: *FDE_AA, FDE_EE*

Applied TDs: *TD0760*

FCS_SNI_EXT.1.1

The TSF shall **use salts that are generated by a DRBG as specified in** FCS_RBG_EXT.1.

FCS_SNI_EXT.1.2

The TSF shall use **unique nonces with a minimum size of [64] bits**.

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: *TSS for FCS_SNI_EXT.1*

6.1.1.25 FCS_VAL_EXT.1/AA Validation

PP Origin: *FDE_AA*

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: *TSS for FCS_VAL_EXT.1/AA*

6.1.1.26 FCS_VAL_EXT.1/EE Validation

PP Origin: *FDE_EE*

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: *TSS for FCS_VAL_EXT.1/EE*

6.1.2 User data protection (FDP)

6.1.2.1 FDP_DSK_EXT.1 Protection of Data on Disk

PP Origin: *FDE_EE*

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: *TSS for FDP_DSK_EXT.1*

6.1.3 Security management (FMT)

6.1.3.1 FMT_MOF.1 Management of Functions Behavior

PP Origin: *FDE_AA*

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: *TSS for FMT_MOF.1*

6.1.3.2 FMT_SMF.1/AA Specification of Management Functions - Authorization Acquisition

PP Origin: *FDE_AA*

Applied TDs: *TD0767*

FMT_SMF.1.1/AA

The TSF shall be capable of performing the following management functions:

a) forwarding requests 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) **configure authorization factors**

TSS Link: *TSS for FMT_SMF.1/AA*

Note: *The term "cryptographically erase" refers to the fact that if a key is protected by using a cryptographic algorithm (e.g. AES-KW), and the key encryption key is zeroized, although the encrypted key is still present in storage it cannot be decrypted as the KEK does not exist anymore.In this particular case, the DEK may be erased by destroying the Key Encryption Key (KEK) that protects it.*

6.1.3.3 FMT_SMF.1/EE Specification of Management Functions - Encryption Engine

PP Origin: *FDE_EE*

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(c), 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: *TSS for FMT_SMF.1/EE*

6.1.3.4 FMT_SMR.1 Security Roles - Authorization Acquisition

PP Origin: *FDE_AA*

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: *TSS for FMT_SMR.1*

6.1.4 Protection of the TSF (FPT)

6.1.4.1 FPT_FUA_EXT.1 Firmware Update Authentication

PP Origin: *FDE_EE*

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: *TSS for FPT_FUA_EXT.1*

6.1.4.2 FPT_KYP_EXT.1/AA Protection of Key and Key Material (AA)

PP Origin: *FDE_AA*

FPT_KYP_EXT.1.1/AA

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: *TSS for FPT_KYP_EXT.1/AA*

6.1.4.3 FPT_KYP_EXT.1/EE Protection of Key and Key Material (EE)

PP Origin: *FDE_EE*

FPT_KYP_EXT.1.1/EE

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.2

TSS Link: *TSS for FPT_KYP_EXT.1/EE*

6.1.4.4 FPT_PWR_EXT.1/AA Power Saving States (AA)

PP Origin: *FDE_AA*

FPT_PWR_EXT.1.1/AA

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

TSS Link: *TSS for FPT_PWR_EXT.1/AA*

6.1.4.5 FPT_PWR_EXT.1/EE Power Saving States (EE)

PP Origin: *FDE_EE*

Applied TDs: *TD0464*

FPT_PWR_EXT.1.1/EE

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

TSS Link: *TSS for FPT_PWR_EXT.1/EE*

6.1.4.6 FPT_PWR_EXT.2 Timing of Power Saving States

PP Origin: *FDE_AA, FDE_EE*

FPT_PWR_EXT.2.1

For each Compliant power saving state defined in *FPT_PWR_EXT.1.1/AA and FPT_PWR_EXT.1.1/EE* ~~FPT_PWR_EXT.1.1~~, the TSF shall enter the Compliant power saving state when the following conditions occur: user-initiated request, **no other conditions**.

TSS Link: *TSS for FPT_PWR_EXT.2*

6.1.4.7 FPT_TST_EXT.1 Testing

PP Origin: *FDE_AA, FDE_EE*

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)**

1. **AES-XTS-128 and AES-XTS-256 encrypt and decrypt**
2. **AES-CBC 256-bit encrypt and decrypt**
3. **CTR_DRBG with AES-256**
4. **ECDSA P-521 with SHA-512 signature verification**
5. **HMAC-SHA-256 MAC generation**
6. **RSA 4096 with SHA-256 signature verification**

TSS Link: *TSS for FPT_TST_EXT.1*

6.1.4.8 FPT_TUD_EXT.1/AA Trusted Update

PP Origin: *FDE_AA*

FPT_TUD_EXT.1.1/AA

The TSF shall provide [authorized users] the ability to query the current version of the TOE **software**.

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: *TSS for FPT_TUD_EXT.1/AA*

6.1.4.9 FPT_TUD_EXT.1/EE Trusted Update

PP Origin: *FDE_EE*

FPT_TUD_EXT.1.1/EE

The TSF shall provide [authorized users] the ability to query the current version of the TOE **software**.

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 a **authenticated firmware update mechanism as described in** FPT_FUA_EXT.1 by the manufacturer prior to installing those updates.

TSS Link: *TSS for FPT_TUD_EXT.1/EE*

6.2 Security Functional Requirements Rationale

The rationale are defined in the documents specified in Section 2 "CC Conformance Claim".

FCS_COP.1(a) has an unresolved dependency on FCS_CKM.1. Signature verification requires the use of preexisting asymmetric public keys; therefore, asymmetric key generation is not required.

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in CC assurance packages.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Table 6: SARs

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	CC	No	No	No	No
	ASE_ECD.1 Extended components definition	CC	No	No	No	No
	ASE_INT.1 ST introduction	CC	No	No	No	No
	ASE_OBJ.1 Security objectives for the operational environment	CC	No	No	No	No
	ASE_REQ.1 Stated security requirements	CC	No	No	No	No
	ASE_SPD.1 Security problem definition	CC	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ASE_TSS.1 TOE summary specification	CC	No	No	No	No
ADV Development	ADV_FSP.1 Basic functional specification	CC	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC	No	No	No	No
ALC Life-cycle support	ALC_CMC.1 Labelling of the TOE	CC	No	No	No	No
	ALC_CMS.1 TOE CM coverage	CC	No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance	CC	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey	CC	No	No	No	No

6.4 Security Assurance Requirements Rationale

The rationale are defined in the documents specified in Section 2 "CC Conformance Claim".

7 TOE Summary Specification

7.1 TOE Security Functionality

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 7: TOE Summary Specification SFR Description

TOE SFRs	Rationale
FCS_AFA_EXT.1, FCS_KDF_EXT.1, FCS_PCC_EXT.1 (Authorization factor, key derivation, PBKDF2)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE supports password authentication factor. Passwords of up to 255 characters are supported and can be comprised of any combination of uppercase characters, lowercase characters, numbers, and any other 8-bit special character. The authentication mechanism consists in deriving the Border Encryption Value (BEV) key (also known as Unlock key) from the user's password, and use that key to unwrap a key (previously wrapped with the correct BEV) using the AES-KW algorithm and the derived BEV. If the unwrapping operation succeeds, it means that the derive BEV is correct, and so is the password. The password is considered validated if the unwrapping function does not return a "Fail" result. The Secure Enclave implements PBKDF2 to derive the BEV key from the user's password. The PBKDF2 is implemented as specified in [SP800-132]☝ following "Option 2b" defined in section 5.4 of the standard. It uses HMAC-SHA-256 as the pseudorandom function (PRF). The input to the PBKDF2 is the 128-bit random salt generated by the TRNG, the user's passcode without any pre-processing, and an iteration count of one. The output is the 256-bit key mentioned above. Next, the output of the PBKDF2 is repeatedly encrypted with the AES-CBC-256 hardware cipher using the 256-bit UID as the encryption key to generate 256 bits of data with each loop iteration. (The UID is described in section 1.5.1.3.) The loop is performed as often as needed to reach a duration between 100 and 150 milliseconds on that device. The output, after all AES iterations have completed, forms the 256-bit Unlock Key or BEV. Note: The number of AES-CBC-256 iterations is calibrated to take at least 100 to 150 milliseconds with a minimum of 50,000 iterations. **Apple silicon** The TOE authenticates the user by unwrapping the class C key stored in the user's keybag with the BEV. **Intel with T2** The TOE authenticates the user by unwrapping the Volume Key with the Media Key first, and then with the BEV (the Volume Key is the DEK and wrapped with both keys).
FCS_AFA_EXT.2 (Authorization factor acquisition)	**Summary** On both Apple silicon and "Intel with T2" Macs, to resume from a compliant power state, the user must reauthenticate to the TOE. The user can reauthenticate using username and password.
FCS_CKM.1(b)	**Summary**

TOE SFRs	Rationale
(Symmetric key generation)	On both Apple silicon and "Intel with T2" Macs, the TOE generates intermediate symmetric keys of 256 bits using the random bit generator specified in FCS_RBG_EXT.1. The keys are generated by the Secure Enclave which invokes internally the TRNG to obtain random bits from the SP800-90A DRBG. The TOE uses these intermediate symmetric keys to protect the key chain from the BEV to the DEK. Keys have 256 bits of security strength.
FCS_CKM.1(c) (Data encryption key generation)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE generates a Data Encryption Key (DEK) of 256 bits using the random bit generator specified in FCS_RBG_EXT.1. The key is generated by the Secure Enclave which invokes internally the TRNG to obtain random bits from the SP800-90A DRBG. The DEK has a security strenght of 256 bits. **Apple silicon** The TOE uses the DEK to encrypt and decrypt data using AES-XTS-256 as described in TSS for FCS_COP.1(f). The DMA storage controller derives a 256-bit tweak and a 256-bit cipher key from this DEK. **Intel with T2** The TOE uses the DEK to encrypt and decrypt data using AES-XTS-128 as described in TSS for FCS_COP.1(f). The DMA storage controller splits the DEK into a 128-bit tweak and a 128-bit cipher key.
FCS_CKM.4(a)/AA, FCS_CKM.4(a)/EE, FCS_CKM.4(b), FCS_CKM.4(d), FCS_CKM_EXT.4(a), FCS_CKM_EXT.4(b), FCS_CKM_EXT.6, FPT_KYP_EXT.1/AA, FPT_KYP_EXT.1/EE (Key destruction)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE leverages NAND flash for non-volatile memory. All symmetric keys that are persistently stored, except for the UID, are cryptographically wrapped and stored in NAND flash. The UID is fused into the SEP's ROM is not accessible by any component outside of the SEP and cannot be erased. 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 erases cryptographic keys and key material from non-volatile memory by performing a single overwrite of zeroes. The TOE leverages DRAM for volatile memory. Keys are stored in volatile memory while being used for their specific operation. Except for the UID and the Unlock Key, all symmetric keys are introduced into volatile memory after being randomly generated or by unwrapping or decrypting a key stored in non-volatile memory. The Unlock Key is introduced into volatile memory after the password-based derivation process has been completed. The TOE will 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/AA and FPT_PWR_EXT.1/EE Keys are only stored in volatile memory when they are required to perform a specific cryptographic operation. Since the keys are being used by the SEP to perform the operation, the SEP tracks the memory location of the key until the operation is complete. Once the keys are no longer required, the key that was used to perform the specific operation is erased from volatile memory by performing a single overwrite of zeroes. The erase operation is performed by the SEP and is not configurable by a user. There are no circumstances that do not conform to the key destruction requirement (e.g. sudden unexpected power loss).

TOE SFRs	Rationale
	The SEP performs the wrapping of keys, which are then sent to the memory controller for storage. The memory controller takes the block of data and the memory location provided by the SEP and stores the data in memory.
FCS_COP.1(a) (Signature verification)	**Summary** On both Apple silicon and "Intel with T2" Macs, signature verification (sigver) is performed as part of the following features: • Installing firmware/software updates ○ TSS for FPT_FUA_EXT.1 and FPT_TUD_EXT.1 • Secure Boot ○ TSS for FPT_TST_EXT.1 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 modules. **Apple silicon** *Algorithm:* ECDSA P-521 sigver *Standard:* FIPS PUB 186-4 *Modules:* • Apple corecrypto Module 13.0 [Apple ARM, User, Software, SL1] • Apple corecrypto Module 13.0 [Apple ARM, Kernel, Software, SL1] • Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2] On Apple silicon Macs, signatures are verified using ECDSA P-521 and SHA-512. The CA public key is embedded in the Mac's Boot ROM code during manufacturing. The TOE image is signed using this key's corresponding private key. **Intel with T2** *Algorithm:* RSA 4096 sigver *Standard:* IEEE 1619 *Modules:* • Apple corecrypto Module 13.0 [Intel, User, Software, SL1] • Apple corecrypto Module 13.0 [Intel, Kernel, Software, SL1] • Apple corecrypto Module 13.0 [Apple ARM, Secure Key Store, Hardware, SL2] On "Intel with T2" Macs, signatures are verified using RSA 4096-bit and SHA-256. The CA public key is embedded in the SEP's Boot ROM code during manufacturing. The TOE image is signed using this key's corresponding private key.
FCS_COP.1(b) (Hash)	**Summary** The TSS for FCS_COP.1(a) describes which hash functions are used and where the hash functions are used. **Apple silicon** *Algorithm:* SHA-512 *Standard:* ISO/IEC 10118-3:2004 *Modules:* • Apple corecrypto Module 13.0 [Apple ARM, User, Software, SL1] • Apple corecrypto Module 13.0 [Apple ARM, Kernel, Software, SL1]

TOE SFRs	Rationale
	• Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2] **Intel with T2** *Algorithm:* SHA-256 *Standard:* ISO/IEC 10118-3:2004 *Modules:* • Apple corecrypto Module 13.0 [Intel, User, Software, SL1] • Apple corecrypto Module 13.0 [Intel, Kernel, Software, SL1] • Apple corecrypto Module 13.0 [Apple ARM, Secure Key Store, Hardware, SL2]
FCS_COP.1(c)/AA, FCS_COP.1(c)/EE (Keyed-hash message authentication)	**Summary** *Algorithm:* HMAC-SHA-256 *Standard:* ISO/IEC 9797-2:2011 On both Apple silicon and "Intel with T2" Macs, the PBKDF2 uses the keyed hash algorithm HMAC-SHA-256 from the corecrypto SKS module as described in the TSS for FCS_PCC_EXT.1. The algorithm supports a key size of 256 bits. **Apple silicon** *Module:* Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2] **Intel with T2** *Module:* Apple corecrypto Module 13.0 [Apple ARM, Secure Key Store, Hardware, SL2]
FCS_COP.1(d) (Key wrapping)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE performs key wrapping using the AES in KW mode according to [SP800-38F]☝. The TOE uses 256-bit keys for this algorithm. AES-KW is an authentication cipher that provides integrity: the decryption operation will only succeed when there is no authentication error. This ensures that the unwrapping operation is performed with the correct key. The TOE uses key wrapping for the following purposes: • Protect the key chain originating from the BEV to the DEK: the TOE uses AES-KW to provide integrity and confidentiality protection of the DEK and intermediate keys. • User authentication: the TOE verifies that the BEV derived from the passcode is able to successfully unwrap the class C key stored in the user's keybag. The failure of unwrapping the user's keybag is also a user authentication failure and therefore access will be denied.
FCS_COP.1(f) (Data encryption and decryption)	**Summary** The TOE uses AES-XTS for data encryption and decryption. The symmetric keys are generated as per the TSS for FCS_CKM.1(b). **Apple silicon** *Algorithm:* AES-XTS-256 *Standard:* IEEE 1619 *Module:* Apple DMA Storage Controller 2.0 [Hardware] On Apple silicon Macs, the TOE supports AES data encryption and decryption in XTS mode using two independent 256-bit keys.

TOE SFRs	Rationale
	Intel with T2 *Algorithm:* AES-XTS-128 *Standard:* IEEE 1619 *Module:* Apple DMA Storage Controller 1.0 [Hardware] On "Intel with T2" Macs, the TOE supports AES data encryption and decryption in XTS mode using two independent 128-bit keys.
FCS_COP.1(g) (Key encryption)	**Summary** *Algorithm:* AES-CBC-256 *Standard:* ISO /IEC 18033-3, ISO/IEC 10116 *Module:* Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2] The TOE uses the SEP's hardware AES-CBC-256 key encryption implementation when generating the Unlock Key as described in the TSS for FCS_PCC_EXT.1. The key size supported is 256 bits.
FCS_KYC_EXT.1, FCS_KYC_EXT.2 (Key chaining initiator & recipient)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE supports BEV sizes of 256 bits. As a key chaining initiator, the TOE maintains a key chain of one, using a submask as the BEV. As a key chaining recipient, the TOE maintains the chain of intermediary keys originating from the BEV to the DEK using the following methods: • Symmetric key generation as specified in FCS_CKM.1(b) • Key wrapping as specified in FCS_COP.1(d) The chain of intermediary keys maintains an effective strength of 256 bits for symmetric keys.
FCS_RBG_EXT.1 (Random bit generation)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE performs deterministic random bit generation services according to NIST SP 800-90A using CTR_DRBG(AES). The DRBG is implemented in hardware and is part of the SEP's TRNG, which is an entropy source based on hardware noise source. The SEP's TRNG consists of a hardware noise source produced by 24 ring oscillators, which produces noise that is collected by a SHA-256 conditioner, which is a vetted conditioning component per NIST SP800-90B. The noise source produces noise constantly and at a higher rate than the process time of the conditioning component, thus ensuring that the amount of entropy input is enough to produce 256 bits with full entropy. The output of the SHA-256 conditioning component is subsequently used by a second conditioning component, the CTR_DRBG mentioned earlier, which is also a vetted conditioning component. The DRBG creates the seeds for initialization and reseeding of the DRBG mechanism. As the seed contains full entropy, the output of the DRBG also provides full entropy. This ensures that the TOE can create symmetric keys with a security strength equal to the key size.
FCS_SNI_EXT.1 (Salt, nonce, and IV generation)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE can generate salts, nonces, and tweaks using the SEP's DRBG. The DRBG is seeded by the SEP's hardware TRNG. Salts are 16 bytes and are used with the PBKDF2. Nonces are 8 bytes and are used with the trusted update process.

TOE SFRs	Rationale
	The AES-CBC initialization vector (IV), used when generating the Unlock Key, is non-repeating and unpredictable. Tweaks are used with the AES-XTS mode of operation. The tweak values should be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer. The tweak value is the physical block number of the media on which the file is being written. This ensures that values cannot be negative. The number is incremented based on the block number values.
FCS_VAL_EXT.1/AA, FCS_VAL_EXT.1/EE (BEV validation)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE will validate a BEV using key wrap as specified in FCS_COP.1(d). The TOE requires the validation of the BEV prior to allowing access to TSF data after exiting a Compliant power saving state. The TOE shall power cycle/reset after 10 consecutive failed validation attempts.
FDP_DSK_EXT.1 (Protection of data on disk)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE provides a dedicated AES-XTS crypto engine built into the Direct Memory Access (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. When the host platform is provisioned at first run, the user is prompted to enable the TOE's embedded FDE encryption management program (FileVault) and enter a username and password. Once enabled, the storage drive of the host platform remains encrypted and protected from unauthorized access; even if the physical storage device is removed and connected to another host platform. The entire storage drive is encrypted with the exception of the following: partition table, Extensible Firmware Interface (EFI) service partition, Apple File System (APFS) container metadata (allocation bitmaps, checkpoint area, EFI jumpstart driver storage, container locker area), recovery volumes, pre-boot volumes, virtual machine (VM) volumes (used by macOS for storing encrypted swap files), and CoreDump partitions (if present). Valid credentials are required to be entered before the drive will be decrypted. If the user does not enable FileVault when provisioning the host platform at first run, FileVault can be enabled later through the System Settings » Privacy & Security menu available via the host platform. By default, the host platform's storage drive is always encrypted. The TOE cryptographic key management changes after enabling FileVault. A recovery key is a randomly generated 28-character code that the user can use to reset their password. The recovery key is generated during the process and manually saved by the user. The recovery key is never stored in the TOE. The recovery key is hashed (SHA-256) and the resulting value is stored in the Secure Enclave. If FileVault is disabled and re-enabled, a new recovery key is generated. See the TSS for FCS_COP.1(f) for details on the AES-XTS implementation.

TOE SFRs	Rationale
FMT_MOF.1 (Function behavior management)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE restricts the ability to modify the behavior of compliant power saving state to authorized users. The TOE requires the user to enter an authorized user's credentials.
FMT_SMF.1/AA, FMT_SMF.1/EE (Management functions)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE supports the following management functions: ● Authorization Acquisition: ○ Forwarding requests to change the DEK to the EE: ➤ The DEK can be changed by starting the Disk Utility and select the appropriate volume to be erased. This forces the TOE to cryptographically erase the DEK and create a new one. Data cannot be recovered after this action. ○ Forwarding requests to cryptographically erase the DEK to the EE: ➤ The DEK can be cryptographically erased by starting the Disk Utility and select the appropriate volume to be erased. ○ Allowing authorized users to change authorization factors or set of authorization factors used: ➤ Once the user successfully authenticates to the TOE, the TOE can be configured to change the authorization factors that can be used: password. ➤ The above can be achieved by navigating to System Settings » Users & Groups » Select the appropriate user » Change Password. ○ Configure authorization factors: ➤ Once the user successfully authenticates to the TOE, the TOE can be configured to change the authorization factors that can be used: password. ➤ The above can be achieved by navigating to System Settings » Users & Groups » Select the appropriate user » Change Password. ● Encryption Engine: ○ Change the DEK: ➤ The DEK can be changed by starting the Disk Utility and select the appropriate volume to be erased. This forces the TOE to cryptographically erase the DEK and create a new one. Data cannot be recovered after this action. ○ Erase the DEK: ➤ The DEK can be cryptographically erased by starting the Disk Utility as an administrator and select the appropriate volume to be erased. ● Authorization Acquisition and Encryption Engine: ○ Initiate TOE firmware/software updates: ➤ The user must successfully login to the TOE before initiating a TOE firmware/software update. After successfully authenticating to the TOE, the user manually downloads the TOE software update(s) from: https://support.apple.com/en-us/HT211683 ➤ Once the update(s) is downloaded, the user needs to initiate the TOE update process by double clicking or right-click » Open the downloaded update.

TOE SFRs	Rationale
	The TOE cryptographically erases the DEK by destroying the keys used to protect it. The concept of the DEK and how erasure is achieved depends on the Mac platform, as described below. **Apple silicon** A Data Encryption Key (DEK) is generated for each file created in an APFS volume. The DEKs are stored in each file metadata within the volume. The DEK is protected by wrapping it with a class C key, which is protected by the BEV key, thus providing data confidentiality based on passcodes. The file metadata where the wrapped DEK is stored is also protected by wrapping it using the Media Key, which provides fast erasure of the data. The Media key is also created when a volume is created or erased, protected by wrapping it with the UID, and stored within the Secure Enclave. When deleting or erasing a volume, the Media Key of the volume is securely deleted (i.e. zeroized) by the Secure Enclave. This causes all DEKs to be cryptographically erased; the wrapped DEKs remain within each file metadata but the Media Key used to protect all file metadata is no longer available. **Intel with T2** A single Data Encryption Key (DEK) is generated for an APFS volume (known as the Volume Key) when a new volume is created or an existing volume is erased. The DEK is stored in the APFS volume. The DEK is protected by wrapping the key with the BEV key, which provides data confidentiality based on passcodes. The wrapped key is wrapped again using the Media Key, which provides fast erasure of the data. The Media key is also created when a volume is created or erased, protected by wrapping it with the UID, and stored within the Secure Enclave. When deleting or erasing a volume, the Media Key of the volume is securely deleted (i.e. zeroized) by the Secure Enclave. This causes the DEK (i.e. Volume Key) to be cryptographically erased; the DEK value remains in the module but the Media Key used to protect it is no longer available.
FMT_SMR.1 (Security roles)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE supports authorized user role and it can associate users to roles.
FPT_FUA_EXT.1, FPT_TUD_EXT.1/AA, FPT_TUD_EXT.1/EE (Software/firmware updates)	**Summary** An Apple server is leveraged for downloading firmware update code packages. The code packages containing the macOS, T2OS/firmware (on Intel with T2 Macs only), and sepOS/firmware are all bundled together as part of the download. The TOE stores the download in a temporary location on flash. Once the download is complete, the SEP verifies (authenticates) the digital signature on the bundle using the RTU public key and the algorithm described in TSS for FCS_COP.1(a). If the verification succeeds, the TOE installs the update and reboots the Mac. If the verification fails, the TOE terminates the update process with an error message. The Mac operating system and software application updates can be downloaded manually through the following website: https://support.apple.com/en-us/HT211683
FPT_PWR_EXT.1/AA, FPT_PWR_EXT.1/EE, FPT_PWR_EXT.2 (Power saving states)	**Summary** On both Apple silicon and "Intel with T2" Macs, the TOE supports the following power savings state: G2(S5) (soft off). The user can either select the menu option Apple menu » Shut Down, or press and hold the physical power button to enter the power saving state.
FPT_TST_EXT.1	**Summary**

TOE SFRs	Rationale
(Testing)	Unless otherwise specified, on both Apple silicon and "Intel with T2" Macs, the TOE performs the following known answer tests (KATs) to verify the correct operation of the cryptographic functions: ● CTR_DRBG with AES-256: The TOE instantiates the DRBG with a known value, invokes the generate function, and compares the generated bits to the expected bits. ● HMAC-SHA-256: MAC generation with a known key and message. ● AES-CBC 256-bit Encrypt/Decrypt: KATs ● Apple silicon: ○ AES-XTS 256-bit Encrypt/Decrypt: this shows the correct operation of AES Encrypt and Decrypt primitive functions with 256-bit keys. ○ ECDSA P-521 with SHA-512 Signature Verification: Satisfied by the Firmware Integrity signature verification test. ● Intel with T2: ○ AES-XTS 128-bit Encrypt/Decrypt: This shows the correct operation of AES Encrypt and Decrypt primitive functions with 128-bit keys. ○ RSA 4096 with SHA-256 Signature Verification: Satisfied by the Firmware Integrity signature verification test. **Apple silicon** During power-up, the application processor loads the Boot ROM which contains the Apple Root CA public key. The Boot ROM authenticates the Low-Level Bootloader (LLB) signature using the Apple Root CA public key. The LLB authenticates system-paired firmware signatures. The LLB authenticates iBoot stage 2 signature. Boot stage 2 authenticates the macOS-paired firmware, Boot Kernel Collection, Auxiliary Kernel Collection (if applicable), system trust cache, and signed system volume signatures. macOS begins execution and authenticates third party kernel extensions (kexts) and OS user space. **Intel with T2** During power-up, the TOE performs a signature verification of firmware and software using the Apple Root CA Public Key. When the Mac is powered-on, the SEP initiates the Secure Boot process. The SEP's Boot ROM first authenticates the signature of the Bridge Boot code (Apple T2 Security Chip Boot ROM code). If the verifications fails, the TOE returns an error and enters the Device Firmware Upgrade (DFU) mode; requiring a correct update to continue. If the verification is successful, the Bridge Boot code then authenticates the signature of the T2 kernel cache. The T2 kernel cache then authenticates the signature of the Unified Extensible Firmware Interface (UEFI) firmware. The UEFI firmware is then used to authenticate the boot.efi file within the Intel processor of the Mac. The boot.efi file then authenticates the macOS immutable kernel. The macOS then authenticates third party kernel extensions (kexts) and OS user space.

8 Abbreviations, Terminology, and References

8.1 Abbreviations

AA
Authorization Acquisition

AES
Advanced Encryption Standard

APFS
Apple File System

API
Application Programming Interface

app
Application

BEV
Border Encryption Value

BIOS
Basic Input/Output System

CA
Certificate Authority

CBC
Cypher Block Chaining

CC
Common Criteria

CEM
Common Evaluation Methodology

cPP
collaborative Protection Profile

CSP
Critical Security Parameters

DAR
Data At Rest

DEK
Data Encryption Key

DFU
Device Firmware Upgrade

DMA
Direct Memory Access

DNS
Domain Name System

DRBG
Deterministic Random Bit Generator

ECDSA
Elliptic Curve Digital Signature Algorithm

EE
Encryption Engine

EFI
Extensible Firmware Interface

FDE
Full Drive Encryption

HMAC
Keyed-hash Message Authentication Code

ISO/IEC
International Organization for Standardization / International Electrotechnical Commission

IV
Initialization Vector

KAT
Known Answer Test

KEK
Key Encryption Key

KW
Key Wrap

MBR
Master Boot Record

NAND
Not AND (inverted boolean AND operation)

OS
Operating System

PII
Personally Identifiable Information

PIN
Personal Identification Number

PBKDF
Password-Based Key Derivation Function

RBG
Random Bit Generator

ROM
Read Only Memory

RSA
Rivest-Shamir-Adleman

RTU
Root of Trust for Update

SAR
Security Assurance Requirement

SED
Self-Encrypting Drive

SEP
Secure Enclave Processor

SFR
Security Functional Requirement

SHA
Secure Hash Algorithm

SKS
Secure Key Store

SoC
System on a Chip

SPI
Serial Peripheral Interface

ST
Security Target

TD
Technical Decision

TLS
Transport Layer Security

TOE
Target of Evaluation

TPM
Trusted Platform Module

TRNG
True Random Number Generator

TSF
TOE Security Functionality

TSS
TOE Summary Specification

UEFI
Unified Extensible Firmware Interface

UID
Unique Identifier

VM
Virtual Machine

VPN
Virtual Private Network

XEX
XOR Encrypt XOR

XTS
XEX Tweakable Block Cipher with Ciphertext Stealing

8.2 References

CC

Common Criteria for Information Technology Security Evaluation
Version 3.1R5
Date April 2017
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf

FDE_AA

collaborative Protection Profile for Full Drive Encryption - Authorization Acquisition
Version 2.0 + Errata 20190201
Date 2019-02-01
Location https://www.niap-ccevs.org/MMO/PP/CPP_FDE_AA_V2.0E.pdf

FDE_EE

collaborative Protection Profile for Full Drive Encryption - Encryption Engine
Version 2.0 + Errata 20190201
Date 2019-02-01
Location https://www.niap-ccevs.org/MMO/PP/CPP_FDE_EE_V2.0E.pdf

SP800-132

Recommendation for Password-Based Key Derivation: Part 1: Storage Applications
Date 2010-12-22
Location https://csrc.nist.gov/pubs/sp/800/132/final

SP800-38F

Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
Date 2012-12-13
Location https://csrc.nist.gov/pubs/sp/800/38/f/final

A Appendixes

A.1 Devices Covered by this Evaluation

Table 8 contains the hardware platforms covered by this evaluation.

For brevity, the processor manufacturer names were left out of the table leaving only the processor names. The Apple silicon SoCs start with the letter M. The Intel® processors start with either Core™ or Xeon®.

The T2 contains the SEP v2.0 core. The T2 micro-architecture (i.e., instruction set architecture) is the following:

- T2: ARMv8.1-A

Table 8 contains the other micro-architectures used in this evaluation in the "microArch" column.

Table 8: Hardware platforms

Marketing Name	Model #	Model Identifier	SoC/Processor	microArch	Security Chip
2023					
MacBook Pro (16-inch, 2023)	A2780	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)	A2779	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)	A2816	Mac14,12	M2 Pro	ARMv8.6-A	SEP v2.0
Mac mini (M2, 2023)	A2686	Mac14,3	M2	ARMv8.6-A	SEP v2.0
2022					
MacBook Pro (13-inch, M2, 2022)	A2338	Mac14,7	M2	ARMv8.6-A	SEP v2.0
MacBook Air (M2, 2022)	A2861	Mac14,2	M2	ARMv8.6-A	SEP v2.0
Mac Studio	A2615	Mac13,2	M1 Ultra	ARMv8.5-A	SEP v2.0
	A2615	Mac13,1	M1 Max	ARMv8.5-A	SEP v2.0
2021					
MacBook Pro (16-inch, 2021)	A2485	MacBookPro18,2	M1 Max	ARMv8.5-A	SEP v2.0
		MacBookPro18,1	M1 Pro	ARMv8.5-A	SEP v2.0
MacBook Pro (14-inch, 2021)	A2442	MacBookPro18,4	M1 Max	ARMv8.5-A	SEP v2.0
		MacBookPro18,3	M1 Pro	ARMv8.5-A	SEP v2.0
iMac (24-inch, M1, 2021)	A2438	iMac21,1	M1	ARMv8.5-A	SEP v2.0
	A2439	iMac21,2	M1	ARMv8.5-A	SEP v2.0
2020					
Mac mini (M1, 2020)	A2348	Macmini9,1	M1	ARMv8.5-A	SEP v2.0
MacBook Air (M1, 2020)	A2337	MacBookAir10,1	M1	ARMv8.5-A	SEP v2.0
MacBook Pro (13-inch, M1, 2020)	A2338	MacBookPro17,1	M1	ARMv8.5-A	SEP v2.0

Marketing Name	Model #	Model Identifier	SoC/Processor	microArch	Security Chip
MacBook Air (Retina, 13-inch, 2020)	A2179	MacBookAir9,1	Core i5-1030NG7 Core i7-1060NG7	Ice Lake	T2
MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports)	A2251	MacBookPro16,2	Core i5-1038NG7 Core i7-1068NG7	Ice Lake	T2
MacBook Pro (13-inch, 2020, Two Thunderbolt 3 ports)	A2289	MacBookPro16,3	Core i5-8257U Core i7-8557U	Coffee Lake	T2
iMac (Retina 5K, 27-inch, 2020)	A2115	iMac20,1 iMac20,2	Core i5-10500 Core i5-10600 Core i7-10700K Core i9-10910	Comet Lake	T2
2019					
MacBook Air (Retina, 13-inch, 2019)	A1932	MacBookAir8,2	Core i5-8210Y	Amber Lake	T2
MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8279U Core i7-8569U	Coffee Lake	T2
MacBook Pro (13-inch, 2019, Two Thunderbolt 3 ports)	A2159	MacBookPro15,4	Core i5-8257U Core i7-8557U	Coffee Lake	T2
MacBook Pro (15-inch, 2019)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-9750H Core i9-9880H Core i9-9980HK	Coffee Lake	T2
MacBook Pro (16-inch, 2019)	A2141	MacBookPro16,1 MacBookPro16,4	Core i7-9750H Core i9-9880H Core i9-9980HK	Coffee Lake	T2
Mac Pro (2019)	A1991	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2
Mac Pro (2019 Rack)	A2304	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2
2018					
MacBook Air (Retina, 13-inch, 2018)	A1932	MacBookAir8,1	Core i5-8210Y	Amber Lake	T2
Mac mini (2018)	A1993	Macmini8,1	Core i5-8500B Core i7-8700B	Coffee Lake	T2
MacBook Pro (15-inch, 2018)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-8750H Core i7-8850H Core i9-8950HK	Coffee Lake	T2
MacBook Pro (13-inch, 2018, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8259U Core i7-8559U	Coffee Lake	T2

Marketing Name	Model #	Model Identifier	SoC/Processor	microArch	Security Chip
2017					
iMac Pro (2017)	A1862	iMacPro1,1	Xeon W-2140B Xeon W-2150B Xeon W-2170B Xeon W-2190B	Skylake	T2

A.2 SFR to CAVP Mapping Table

The CAVP certificates contain several different SoCs and micro-architectures in the operational environment (OE). The relationship between the SoCs and micro-architectures used by the devices claimed in this evaluation are specified in Appendix A.1.

The following convention has been used in the tables of this appendix to identify the cryptographic modules.

DMA

- Apple DMA Storage Controller 2.0 [Hardware], and/or
- Apple DMA Storage Controller 1.0 [Hardware]

KRN

- Apple corecrypto Module 13.0 [Apple ARM, Kernel, Software, SL1], and/or
- Apple corecrypto Module 13.0 [Intel, Kernel, Software, SL1]

SEP

- SEP Hardware v2.0 in Apple silicon, and/or
- SEP Hardware v2.0 in Apple T2

SKS

- Apple corecrypto Module 13.0 [Apple silicon, Secure Key Store, Hardware, SL2]

USR

- Apple corecrypto Module 13.0 [Apple ARM, User, Software, SL1], and/or
- Apple corecrypto Module 13.0 [Intel, User, Software, SL1]

The T2 is marketed as Apple ARM technology and runs T2OS 13.

The DMA module cannot be tested through the CAVP, therefore a compliance test accepted by NIAP has been used for verifying the correctness of the algorithms implemented.

Table 9: Cryptographic algorithm table

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
FCS_COP.1(a) Signature verification	ECDSA SigVer [FIPS 186-4]	Curve: P-521 with SHA-512 (Apple silicon)	USR	vng_ltc	A3488
			KRN	vng_ltc	A3521
			SKS	vng_ltc	A4259
	RSA SigVer [FIPS 186-4]	Modulo: 4096 with SHA-256 PKCS 1.5 and PKCSPSS (Intel)	USR	c_avx2	A3506
			KRN	c_avx2	A3623
		Modulo: 4096 with SHA-256 PKCS 1.5 and PKCSPSS (T2)	SKS	vng_ltc	A4109
FCS_COP.1(b) Hash	SHS Byte-oriented mode [FIPS 186-4]	SHA2-512 (Apple silicon)	USR	vng_ltc	A3488
			KRN	vng_ltc	A3521
			SKS	vng_ltc	A4259
	SHS	SHA2-256	USR	vng_intel	A3512

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
	Byte-oriented mode [FIPS 186-4]	(Intel)	KRN	vng_intel	A3628
		SHA2-256 (T2)	SKS	vng_neon	A4110
FCS_COP.1(c)/AA, FCS_COP.1(c)/EE Keyed hash	HMAC Byte-oriented mode [ISO/IEC 9797-2:2011]	HMAC-SHA2-256 (Apple silicon)	SKS	vng_neon	A4260
		HMAC-SHA2-256 (T2)	SKS	vng_neon	A4110
FCS_COP.1(d) Key wrapping	AES [FIPS 197]	KW 256 bit encrypt, decrypt (Apple silicon) [SP800-38F]	SKS	c_asm	A4254
		KW 256 bit encrypt, decrypt (T2) [SP800-38F]	SKS	c_asm	A4104
FCS_COP.1(f) Data encrypt/decrypt	AES [FIPS 197]	XTS 256 bit encrypt, decrypt (Apple silicon) [SP800-38F]	DMA	n/a	None (verified through compliance test accepted by NIAP)
		XTS 128 bit encrypt, decrypt (Intel/T2) [SP800-38F]	DMA	n/a	None (verified through compliance test accepted by NIAP)
FCS_COP.1(g) Key encryption	AES [ISO/IEC 18033-3]	CBC 256 bit encrypt [ISO/IEC 10116]	SEP	M2 Max (skg)	A3496
				M2 Pro (skg)	A3496
				M2 (skg)	A3496
				M1 Ultra (skg)	A3496

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
				M1 Max (skg)	A3496
				M1 Pro (skg)	A3496
				M1 (skg)	A1469
				T2 (skg)	C330
		CBC 256 bit encrypt, decrypt [ISO/IEC 10116]	SKS	M2 Max (c_asm)	A4254
				M2 Pro (c_asm)	A4254
				M2 (c_asm)	A4254
				M1 Ultra (c_asm)	A4254
				M1 Max (c_asm)	A4254
				M1 Pro (c_asm)	A4254
				M1 (c_asm)	A4254
				T2 (c_asm)	A4104
FCS_RBG_EXT.1 Random bit generation	CTR_DRBG [SP800-90A]	AES-256	SEP	M2 Max (trng)	A3490
				M2 Pro (trng)	A3490
				M2 (trng)	A3490
				M1 Ultra (trng)	A3490
				M1 Max (trng)	A3490
				M1 Pro (trng)	A3490
				M1 (trng)	A1362
				T2	DRBG 2029

The following table shows the full coverage of CAVP tests for the Apple silicon models used in the devices covered by this evaluation and specified in Appendix A.1.

Table 10: Coverage of CAVP certificates for Apple silicon

SoC	Micro Architecture	USR		KRN		SKS					SEP (v2.0)	
		ECDSA SigVer	SHA-512	ECDSA SigVer	SHA-512	ECDSA SigVer	SHA-512	HMAC SHA-256	AES-CBC	AES-KW	AES-CBC	CTR_DRBG
M1	ARMv8.5-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A1469	A1362
M1 Max	ARMv8.5-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490
M1 Pro	ARMv8.5-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490
M1 Ultra	ARMv8.5-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490
M2	ARMv8.6-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490
M2 Max	ARMv8.6-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490
M2 Pro	ARMv8.6-A	A3488	A3488	A3521	A3521	A4259	A4259	A4260	A4254	A4254	A3496	A3490

The following table shows the coverage of CAVP tests for the Intel processors used in the devices covered by this evaluation and specified in Appendix A.1. For those processor models not tested, the last column indicates the equivalent processor on which the CAVP tests were performed. The equivalence argument for these processors is that the reference testing is performed on a processor of the same Intel Micro Architecture and Intel processor Generation.

Table 11: Coverage of CAVP certificates for Intel Processors

Processor	Gen	Micro Architecture	USR		KRN		Equivalent processor
			RSA SigVer	SHA-256	RSA SigVer	SHA-256	
Intel Xeon W-2140B	W	Skylake	A3506	A3512	A3623	A3628	Tested
Intel Xeon W-2150B	W	Skylake					Intel Xeon W-2140B
Intel Xeon W-2170B	W	Skylake					Intel Xeon W-2140B
Intel Xeon W-2190B	W	Skylake					Intel Xeon W-2140B
Intel Xeon W-3223	W	Cascade Lake	A3506	A3512	A3623	A3628	Tested
Intel Xeon W-3235	W	Cascade Lake					Intel Xeon W-3223
Intel Xeon W-3245	W	Cascade Lake					Intel Xeon W-3223
Intel Xeon W-3265	W	Cascade Lake					Intel Xeon W-3223
Intel Xeon W-3265M	W	Cascade Lake					Intel Xeon W-3223
Intel Xeon W-3275M	W	Cascade Lake					Intel Xeon W-3223
Intel Core i5-8210Y	8th	Amber Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i5-8257U	8th	Coffee Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i5-8259U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i5-8279U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i7-8557U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i7-8559U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i7-8569U	8th	Coffee Lake					Intel Core i5-8257U
Intel Core i5-8500B	8th	Coffee Lake					Intel Core i7-8700B
Intel Core i7-8700B	8th	Coffee Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i7-8750H	8th	Coffee Lake					Intel Core i7-8700B
Intel Core i7-8850H	8th	Coffee Lake					Intel Core i7-8700B
Intel Core i9-8950HK	8th	Coffee Lake					Intel Core i7-8700B
Intel Core i7-9750H	9th	Coffee Lake					
Intel Core i9-9880H	9th	Coffee Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i9-9880HK	9th	Coffee Lake					
Intel Core i5-10500	10th	Comet Lake					Intel Core i7-10700K
Intel Core i5-10600	10th	Comet Lake					Intel Core i7-10700K
Intel Core i7-10700K	10th	Comet Lake	A3506	A3512	A3623	A3628	Tested

Processor	Gen	Micro Architecture	USR		KRN		Equivalent processor
			RSA SigVer	SHA-256	RSA SigVer	SHA-256	
Intel Core i9-10910	10th	Comet Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i5-1030NG7	10th	Ice Lake					Intel Core i7-1060NG7
Intel Core i5-1038NG7	10th	Ice Lake					Intel Core i7-1060NG7
Intel Core i7-1060NG7	10th	Ice Lake	A3506	A3512	A3623	A3628	Tested
Intel Core i7-1068NG7	10th	Ice Lake					Intel Core i7-1060NG7

The following table shows the full coverage of CAVP tests for the Apple T2 Security Chip, used as the security chip in devices using Intel processors, as specified in Appendix A.1.

Table 12: Coverage of CAVP certificates for Apple T2 Security Chip

SoC	Micro Architecture	SKS					SEP	
		RSA SigVer	SHA-256	HMAC SHA-256	AES-CBC	AES-KW	AES-CBC	CTR_DRBG
T2	ARMv8.1-A	A4109	A4110	A4110	C330	A4104	C330	DRBG 2029